

# **SKILL SS01: SOFTWARE LIBRE**



## **Test Project día 2**

Administración de sistemas Linux

**Madrid 2013**

## Índice de contenido

1 Instrucciones generales.....	3
1.1 Observaciones.....	3
1.2 Equipamiento.....	4
1.3 Distribución a utilizar.....	4
1.4 Material de apoyo.....	4
2 Test Project.....	5
2.1 Ejecicios .....	5
2.2 Descripción del Ejercicio 1.....	5
2.3 Descripción del Ejercicio 2.....	5
3 Esquema de puntuación.....	7
4 Temporalización del ejercicio.....	8

---

# 1 Instrucciones generales

---

## 1.1 Observaciones

- Ponga especial atención a lo que se pide en cada uno de los apartados de la prueba.
- Mantener la mesa ordenada y recogida.
- El competidor sólo podrá disponer de un bolígrafo y de la documentación que le facilita la organización.
- Los móviles deben estar apagados.
- No se permite el uso de ningún servicio de mensajería instantánea o de comunicación interpersonal (correo electrónico, chats, foros de Internet, etc).
- El competidor recibirá, junto con este Test Project Final, una hoja con las imágenes ampliadas del sistema informático a implantar, para que pueda realizar la prueba con mayor facilidad.
- Se deben respetar los nombres, IPs, usuarios, contraseñas, configuraciones, etc. indicados en el presente Test Project (TP) y descartar los que puedan traer los sistemas en la instalación por defecto.
- Aquello que no esté descrito de una forma explícita en el presente TP queda a la completa discreción del competidor para su configuración o nombramiento.
- Cuando el competidor finalice la prueba debe apagar todos los ordenadores, reales y virtuales. La corrección comenzará encendiendo los equipos y comprobando que tras un apagado, todo funciona correctamente.

## **1.2 Equipamiento**

Para la realización de la prueba se tiene disponible el siguiente equipamiento:

- 1 ordenador con sistema operativo Ubuntu Desktop instalado, para ser usado como ordenador cliente. Su hostname será asterix.
- 1 ordenador con sistema operativo Ubuntu 12.04 instalado para ser usado como ordenador servidor. Su hostname será simpsons.
- 1 switch de red.
- 2 Latiguillos de red categoría 5/6.
- Repositorio de paquetes necesarios para la realización de la práctica.
- Una llave USB conteniendo:
  - archivo llamado `authorized_keys.basicas` con las llaves públicas ssh a usar en el ejercicio 1.
  - el contenido del directorio `/var/lib/puppet/reports` con datos ficticios para la realización del ejercicio 2.

## **1.3 Distribución a utilizar**

Para el desarrollo de la prueba se va a utilizar la distribución Ubuntu 12.04 en sus versiones “Server” y “Desktop”.

## **1.4 Material de apoyo**

En la llave USB que se entrega al concursante hay un directorio llamado Documentación con manuales y tutoriales sobre los servicios y tecnologías a usar en la práctica.

## 2 Test Project

---

### 2.1 Ejecicios

El test correspondiente al día 2 constará de los siguientes ejercicios:

1. Instalación de un servidor puppetmaster donde se crearán reglas de puppet que deberán ejecutar los clientes.
2. Realización de un script en Python, Perl o Bash que elabore un informe diario sobre el funcionamiento del servidor de puppet.

### 2.2 Descripción del Ejercicio 1

El concursante deberá realizar la instalación y configuración completa de un servidor puppetmaster y un servidor horario ntp. Para ello empleará el ordenador servidor de la práctica anterior, es decir: simpsons.

El concursante deberá elaborar e instalar en el servidor reglas que obligue a los clientes a:

- Regla pkgsync.pp: ejecutar pkgsync al menos una vez al día.
- Regla paquete\_ntp.pp: garantizar que el paquete ntp está siempre instalado.
- Regla configura\_ntp.pp: garantizar la existencia del texto:  

```
NTPSERVERS="ip_del_servidor"
```

en el fichero /etc/default/ntp-servers, incluso aunque el usuario del cliente elimine o modifique esas líneas.
- Regla entrada\_ssh.pp: garantizar que el fichero /root/.ssh/authorized\_keys contiene la clave ssh pública contenida en un fichero que el jurado suministrará. Esta clave debe estar siempre en ese fichero, sin perjuicio de que se puedan añadir otras claves a él.

Además, el concursante deberá configurar correctamente como cliente de puppet el ordenador asterix y probar que obedece las instrucciones del servidor.

### 2.3 Descripción del Ejercicio 2

El concursante deberá realizar un script en Python, Perl o Bash que realice las

siguientes funciones de forma automática una vez al día:

- Recorrer el directorio `/var/lib/puppet/reports` del servidor de puppet, extrayendo la fecha y hora de la última conexión correcta de los clientes del servidor. La fecha y hora deberá mostrarla por consola según va realizando el recorrido del directorio.
- Crear el fichero `errores_puppet.txt` que contenga un listado ordenado cronológicamente de los clientes de puppet que no se hayan conectado correctamente en el último mes.
- Enviar mediante correo electrónico (desde una cuenta de este ordenador y mediante postfix), ese listado ordenado a la dirección de correo electrónico: [jurado@asterix.dibujos.local](mailto:jurado@asterix.dibujos.local)

Previamente, y para el correcto funcionamiento del script, el concursante deberá realizar también estas tareas:

- Configurar postfix en este ordenador para que permita su uso como servidor de correo.
- Crear una cuenta de usuario en ese ordenador llamada jurado.

Nota importante: Para la realización del ejercicio 2 no es necesario que se haya completado el ejercicio 1. Se deben usar los datos ficticios del directorio `/var/lib/puppet/reports` proporcionados por el jurado del skill.

### 3 Esquema de puntuación

---

La validación de la prueba atenderá a criterios objetivos, en términos de apto/no apto, funciona/no funciona o correcto/incorrecto. En el caso del segundo ejercicio se establecen además dos criterios subjetivos para los que se explicita aquí la máxima puntuación que le podrán otorgar los jurados de la prueba.

Cada criterio global de calificación se desglosará en items.

Los criterios objetivos de calificación para esta prueba son:

- |   |                    |
|---|--------------------|
| 1.- Instalación correcta de puppet y puppetmaster         | (Total: 4 puntos)  |
| Instalación de paquetes necesarios                        | 1 punto            |
| Creación clase de prueba en el servidor                   | 1 punto            |
| Gestión de los certificados ssl entre cliente y servidor  | 2 puntos           |
| 2.- Creación de reglas de puppet (a probar en el cliente) | (Total: 9 puntos)  |
| Correcto funcionamiento de la regla pkgsync.pp            | 2 puntos           |
| Configuración correcta del servicio ntp                   | 1 punto            |
| Funcionamiento correcto de la regla paquete_ntp.pp        | 2 puntos           |
| Funcionamiento correcto de la regla configura_ntp.pp      | 2 puntos           |
| Funcionamiento correcto de la regla entrada_ssh.pp        | 2 puntos           |
| 3.- Generación del fichero errores_puppet.txt             | (Total: 12 puntos) |
| Mostrar correctamente la fecha y hora extraída            | 3 puntos           |
| Creación del fichero errores_puppet.txt                   | 0,5 puntos         |
| Listado ordenado en el fichero                            | 2 puntos           |
| Configuración correcta del servidor postfix               | 2 puntos           |
| Creación de cuenta de usuario "jurado"                    | 0,5 puntos         |
| Envío correcto del correo con el contenido del fichero    | 1 punto            |

Los criterios subjetivos de calificación para esta prueba son:

Comentarios y documentación del script	1 punto
Elegancia del algoritmo usado	2 puntos

## **4 Temporalización del ejercicio**

---

La prueba tiene un tiempo determinado de seis horas. El concursante debe decidir la mejor manera de planificar su tiempo.