

07 Debconf

Con el paso del tiempo, ciertas utilidades se convierten en componentes esenciales para lograr una automatización confiable y reproducible de instalaciones en entornos GNU/Linux, especialmente en Debian y derivados. Una de estas herramientas clave es **Debconf**, que facilita la preconfiguración de paquetes y evita la intervención del usuario durante la instalación.

Debconf

Debconf es un sistema de configuración que utilizan los paquetes de Debian para interactuar con el usuario durante la instalación. A través de él, los mantenedores de paquetes definen una serie de preguntas (como contraseñas, rutas de instalación, opciones de activación de servicios, entre otras) que el sistema solicitaría en una instalación manual.

Sin embargo, en entornos donde se busca automatizar el despliegue (servidores, cloud, laboratorios docentes, infraestructuras reproducibles con Ansible o Puppet), resulta mucho más eficiente establecer respuestas de antemano para evitar la intervención manual.

Debconf permite:

- **Preconfigurar paquetes** antes de su instalación.
- **Consultar valores actuales** definidos durante la instalación.
- **Exportar y reutilizar configuraciones** en múltiples máquinas, garantizando consistencia.

Para poder operar con estas opciones es necesario instalar el paquete de utilidades:

```
root@serverPrueba:~# apt update
root@serverPrueba:~# apt install -y debconf-utils
```

Este paquete incluye comandos muy útiles como:

- **debconf-get-selections**: muestra las opciones de configuración elegidas para los paquetes instalados en el sistema.
- **debconf-set-selections**: permite asignar valores previamente guardados a paquetes, evitando que el instalador interactúe con el usuario.

Ejemplo

Ver opciones configuradas en el paquete *openssh-server*.

```
root@serverPrueba:~# debconf-get-selections | grep ^openssh
openssh-server  openssh-server/password-authentication  boolean true
openssh-server  openssh-server/permit-root-login        boolean true
```

En el ejemplo anterior, se observan parámetros definidos automáticamente para el paquete *openssh-server*:

- Se permite la autenticación por contraseña.
- Se autoriza el inicio de sesión como usuario root.

Estos valores pueden:

1. Guardarse en un **fichero de configuración**.
2. Transferirse a un sistema remoto.
3. Aplicarse con el comando:

```
debconf-set-selections < fichero_opciones
```

Esto garantiza que, cuando se ejecute posteriormente una instalación con **apt**, los paquetes ya tengan definidas sus opciones sin requerir interacción manual.

Integración en procesos de automatización

Debconf se integra fácilmente en flujos de trabajo con herramientas como **Ansible**, **Puppet** o **Chef**, donde el administrador busca:

- Ejecutar tareas **de manera no interactiva**.
- Asegurar que todos los servidores tengan **la misma configuración inicial**.
- Reducir errores humanos y acelerar despliegues masivos.

En el caso concreto de Ansible, el preajuste con **debconf-set-selections** se combina con módulos de gestión de paquetes (**apt**) para realizar instalaciones controladas sin intervención manual.

Beneficios clave

- Homogeneidad en entornos con muchos servidores.
- Reducción del tiempo necesario para desplegar servicios.
- Posibilidad de auditar y versionar configuraciones (almacenándolas en repositorios junto con el resto de la infraestructura como código).
- Flexibilidad para reconfigurar paquetes de manera centralizada.

Alternativa para otras distribuciones

En distribuciones como Red Hat, CentOS o Rocky Linux la herramienta principal para la automatización de instalaciones es **Kickstart**, que funciona como alternativa a **Debconf** en Debian.

Kickstart permite crear un archivo de configuración (usualmente llamado **ks.cfg**) donde se especifican todos los parámetros necesarios para realizar una instalación completamente desatendida, incluyendo:

- Particionado de discos
- Configuración de red
- Selección de paquetes a instalar
- Configuración de usuarios y contraseñas
- Parámetros de arranque y otras opciones

Instalación de slapd

Slapd en Linux es el demonio o servicio del servidor LDAP (Lightweight Directory Access Protocol). Es un programa que se ejecuta en segundo plano y gestiona un directorio centralizado para almacenar, buscar y administrar información, como usuarios, permisos y configuraciones en la red.

Slapd forma parte del paquete OpenLDAP y es responsable de manejar las peticiones de clientes LDAP, permitiendo autenticación, control de acceso y replicación de datos para entornos que requieren un directorio de servicios confiable y escalable.

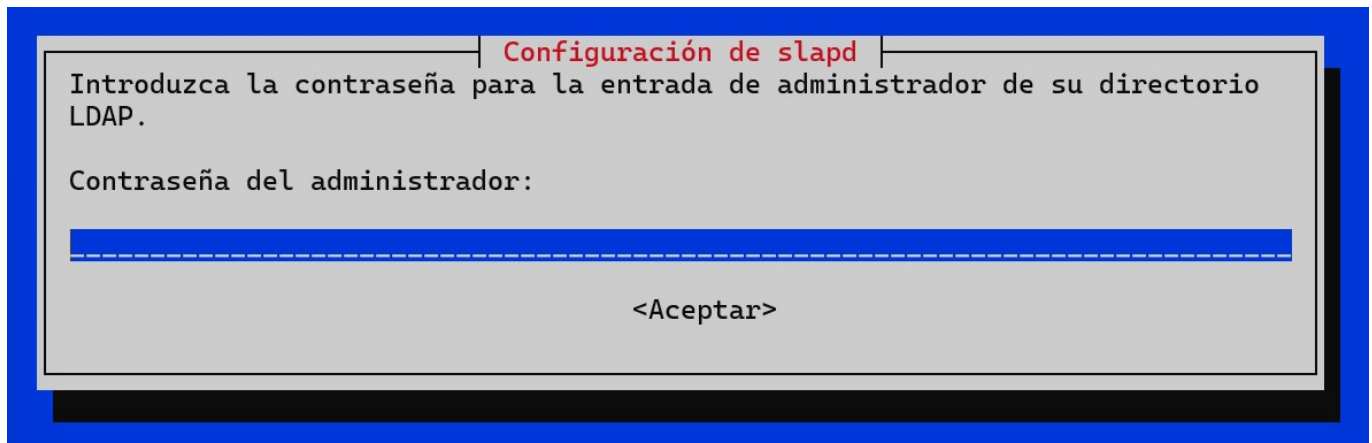
Especialmente interesante va a ser automatizar la instalación de slapd ya que es un software que realiza preguntas de configuración por lo que su automatización se ha de realizar haciendo uso de Debconf en una máquina Debian.

Proceso de instalación de slapd

Se recomienda hacer uso de una máquina de prueba en la que instalamos el servicio para así poder tener las respuestas generadas.

A la hora de realizar el proceso de instalación de *slapd* desde línea de comandos ocurre que nos realizan preguntas de configuración como es el caso de la contraseña. Esto impide realizar el proceso de instalación desde un playbook directamente como hemos visto hasta ahora por lo que tenemos que saber las respuestas de configuración mediante el comando *debconf-get-selections*.

```
root@serverPrueba:~# apt install slapd -y
```



```
root@serverPrueba:~# debconf-get-selections | grep ^slapd
slapd  shared/organization      string  nodomain
slapd  slapd/domain                 string  nodomain
slapd  slapd/dump_database          select  when needed
slapd  slapd/dump_database_destdir   string  /var/backups/slapd-VERSION
slapd  slapd/internal/adminpw        password
slapd  slapd/internal/generated_adminpw password
slapd  slapd/invalid_config          boolean true
slapd  slapd/move_old_database        boolean true
slapd  slapd/no_configuration         boolean false
```

```
slapd    slapd/password1 password
slapd    slapd/password2 password
slapd    slapd/password_mismatch note
slapd    slapd/postinst_error    note
slapd    slapd/purge_database    boolean false
slapd    slapd/upgrade_slapcat_failure error
```

Cada línea de la salida de `debconf-get-selections` está formada por cuatro campos, cuyos significados son:

- **Nombre del paquete:** Especifica a qué paquete corresponde la configuración (en el ejemplo, siempre es `slapd`). Siempre será el nombre del paquete gestionado por debconf, como `slapd`, `mysql-server`, etc.
- **Nombre de la pregunta (o clave):** Es el identificador de la opción o ajuste concreto al que se refiere (por ejemplo, `slapd/domain`, `shared/organization`). Es el identificador específico de configuración dentro del paquete, como `slapd/domain` para el dominio LDAP.
- **Tipo de valor:** Indica el tipo de información esperada, como `string` (cadena de texto), `password` (contraseña), `boolean` (valor verdadero/falso), `select` (opción entre varias) o `note`, `error` (utilizados para avisos y errores). Define el tipo de dato que se almacena, como:
 - `string` para textos.
 - `password` para contraseñas que se ocultan.
 - `boolean` para opciones sí/no (`true/false`).
 - `select` para seleccionar entre varias opciones.
 - `note` y `error` para mensajes informativos o de error.
- **Valor configurado:** Es el valor actual usado para esa clave y tipo, por ejemplo, "nodomain", "true", "when needed" o una ruta de directorio.

En estos momentos para realizar la automatización es necesario tener estas respuestas en el *NodeControler*, para hacer este proceso podemos hacer uso de SCP para transferir el archivo.

```
root@serverPrueba:~# debconf-get-selections | grep ^slapd >
/tmp/respuestas_slapd.txt
```

Transferimos de la máquina de pruebas al *NodeControler*.

```
ansible@debian64:~/practicas/pr04/slapd$ scp
ansible@192.168.10.5:/tmp/respuestas_slapd.txt /tmp
respuestas_slapd.txt
```

Proceso de automatización mediante playbook

El proceso de automatización mediante playbook de una instalación como esta puede realizarse de varias formas, en este manual mostraremos dos siendo la segura la forma preferible.

Primera forma de instalar el servicio slapd

```

---

- name: Instalación de slapd mediante módulo shell
  hosts: srv1

  tasks:
    - name: Configurar preguntas de instalación
      ansible.builtin.shell:
        cmd: debconf-set-selections
        stdin: |
          slapd shared/organization string nodomain
          slapd slapd/domain string nodomain
          slapd slapd/dump_database select when needed
          slapd slapd/dump_database_destdir string /var/backups/slapd-VERSION
          slapd slapd/internal/adminpw password abc123.
          slapd slapd/internal/generated_adminpw password abc123.
          slapd slapd/invalid_config boolean true
          slapd slapd/move_old_database boolean true
          slapd slapd/no_configuration boolean false
          slapd slapd/password1 password
          slapd slapd/password2 password
          slapd slapd/password_mismatch note
          slapd slapd/postinst_error note
          slapd slapd/purge_database boolean false
          slapd slapd/upgrade_slapcat_failure error

    - name: Instalación del slapd
      ansible.builtin.apt:
        name: slapd
        state: present
        update_cache: yes

```

```

ansible@debian64:~/practicass/pr04/slapd$ ansible-playbook slapd.yaml

```

```

PLAY [Instalación de slapd mediante módulo shell]

```

```

*****
*****

```

```

TASK [Gathering Facts]

```

```

*****
*****

```

```

ok: [srv1]

```

```

TASK [Configurar preguntas de instalación]

```

```

*****
*****

```

```

changed: [srv1]

```

```

TASK [Instalación del slapd]

```

```

*****

```

```
*****
changed: [srv1]

PLAY RECAP
*****
*****
srv1      : ok=3    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Podemos comprobar que está instalado.

```
ansible@debian64:~$ systemctl status slapd
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
   Loaded: loaded (/etc/init.d/slapd; generated)
   Drop-In: /usr/lib/systemd/system/slapd.service.d
            └─slapd-remain-after-exit.conf
   Active: active (running) since Thu 2025-09-25 18:55:53 CEST; 5min ago
     Docs: man:systemd-sysv-generator(8)
   Process: 382613 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCCESS)
     Tasks: 3 (limit: 1108)
    Memory: 5.3M
       CPU: 26ms
    CGroup: /system.slice/slapd.service
            └─382619 /usr/sbin/slapd -h "ldap:/// ldapi:///" -g openldap -u openldap -F /etc/ldap/slapd.d
```

Nota: Importante tener en cuenta que los valores de las respuestas para *debconf-set-selections* tienen que estar separados por espacio, no por tabulador.

Segunda forma de instalar el servicio slapd mediante el módulo

Documentación del módulo debconf

En Ansible existe un módulo específico para gestionar respuestas de debconf en Debian es `ansible.builtin.debconf`.

Este módulo permite configurar paquetes Debian mediante la herramienta *debconf-set-selections*, respondiendo preguntas durante la instalación o preconfigurando opciones de paquetes. Como parámetros clave del módulo tenemos los siguientes:

- **name:** nombre del paquete a configurar.
- **question:** la clave de la pregunta de debconf.
- **value:** valor que se quiere asignar.
- **vtype:** tipo de valor (string, select, password, boolean, etc.).

Es importante tener en cuenta que con este módulo no es necesario preconfigurar una tarea por cada pregunta para la instalación o configuración, es decir, podemos configurar solamente una tarea por cada

pregunta relevante siendo estas las de tipo string, select, password o boolean que afectan opciones concretas del paquete.

Por lo tanto una propuesta de playbook para la instalación del servicio sería el siguiente:

```
---
- name: Instalar y configurar slapd mediante módulo debconf
  hosts: srv1

  tasks:
    - name: Configurar shared/organization
      ansible.builtin.debconf:
        name: slapd
        question: shared/organization
        vtype: string
        value: nodomain

    - name: Configurar slapd/domain
      ansible.builtin.debconf:
        name: slapd
        question: slapd/domain
        vtype: string
        value: nodomain

    - name: Configurar slapd/dump_database
      ansible.builtin.debconf:
        name: slapd
        question: slapd/dump_database
        vtype: select
        value: when needed

    - name: Configurar slapd/dump_database_destdir
      ansible.builtin.debconf:
        name: slapd
        question: slapd/dump_database_destdir
        vtype: string
        value: /var/backups/slapd-VERSION

    - name: Configurar slapd/internal/adminpw
      ansible.builtin.debconf:
        name: slapd
        question: slapd/internal/adminpw
        vtype: password
        value: abc123.

    - name: Configurar slapd/internal/generated_adminpw
      ansible.builtin.debconf:
        name: slapd
        question: slapd/internal/generated_adminpw
        vtype: password
        value: abc123.
```

```
- name: Configurar slapd/invalid_config
  ansible.builtin.debconf:
    name: slapd
    question: slapd/invalid_config
    vtype: boolean
    value: true

- name: Configurar slapd/move_old_database
  ansible.builtin.debconf:
    name: slapd
    question: slapd/move_old_database
    vtype: boolean
    value: true

- name: Configurar slapd/no_configuration
  ansible.builtin.debconf:
    name: slapd
    question: slapd/no_configuration
    vtype: boolean
    value: false

- name: Configurar slapd/password1
  ansible.builtin.debconf:
    name: slapd
    question: slapd/password1
    vtype: boolean
    value: ""

- name: Configurar slapd/password2
  ansible.builtin.debconf:
    name: slapd
    question: slapd/password2
    vtype: boolean
    value: ""

- name: Configurar slapd/password_mismatch
  ansible.builtin.debconf:
    name: slapd
    question: slapd/password_mismatch
    vtype: note
    value: ""

- name: Configurar slapd/postinst_error
  ansible.builtin.debconf:
    name: slapd
    question: slapd/postinst_error
    vtype: note
    value: ""

- name: Configurar slapd/purge_database
  ansible.builtin.debconf:
    name: slapd
    question: slapd/purge_database
    vtype: boolean
```



```

    value: false

- name: Configurar slapd/upgrade_slapcat_failure
  ansible.builtin.debconf:
    name: slapd
    question: slapd/upgrade_slapcat_failure
    vtype: error
    value: ""

- name: Instalar paquete slapd
  ansible.builtin.apt:
    name: slapd
    state: present
    update_cache: yes

# slapd shared/organization string nodomain
# slapd slapd/domain string nodomain
# slapd slapd/dump_database select when needed
# slapd slapd/dump_database_dest_dir string /var/backups/slapd-VERSION
# slapd slapd/internal/adminpw password abc123.
# slapd slapd/internal/generated_adminpw password abc123.
# slapd slapd/invalid_config boolean true
# slapd slapd/move_old_database boolean true
# slapd slapd/no_configuration boolean false
# slapd slapd/password1 password
# slapd slapd/password2 password
# slapd slapd/password_mismatch note
# slapd slapd/postinst_error note
# slapd slapd/purge_database boolean false
# slapd slapd/upgrade_slapcat_failure error

```

```
---
```

```

- name: Instalar y configurar slapd mediante módulo debconf
  hosts: srv1

```

```
tasks:
```

```

- name: Configurar shared/organization
  ansible.builtin.debconf:
    name: slapd
    question: shared/organization
    vtype: string
    value: nodomain

- name: Configurar slapd/domain
  ansible.builtin.debconf:
    name: slapd
    question: slapd/domain
    vtype: string
    value: nodomain

- name: Configurar slapd/dump_database

```

```
ansible.builtin.debconf:
  name: slapd
  question: slapd/dump_database
  vtype: select
  value: when needed

- name: Configurar slapd/dump_database_destdir
  ansible.builtin.debconf:
    name: slapd
    question: slapd/dump_database_destdir
    vtype: string
    value: /var/backups/slapd-VERSION

- name: Configurar slapd/internal/adminpw
  ansible.builtin.debconf:
    name: slapd
    question: slapd/internal/adminpw
    vtype: password
    value: abc123.

- name: Configurar slapd/internal/generated_adminpw
  ansible.builtin.debconf:
    name: slapd
    question: slapd/internal/generated_adminpw
    vtype: password
    value: abc123.

- name: Configurar slapd/invalid_config
  ansible.builtin.debconf:
    name: slapd
    question: slapd/invalid_config
    vtype: boolean
    value: true

- name: Configurar slapd/move_old_database
  ansible.builtin.debconf:
    name: slapd
    question: slapd/move_old_database
    vtype: boolean
    value: true

- name: Configurar slapd/no_configuration
  ansible.builtin.debconf:
    name: slapd
    question: slapd/no_configuration
    vtype: boolean
    value: false

- name: Configurar slapd/password1
  ansible.builtin.debconf:
    name: slapd
    question: slapd/password1
    vtype: boolean
    value: ""
```

```
- name: Configurar slapd/password2
  ansible.builtin.debconf:
    name: slapd
    question: slapd/password2
    vtype: boolean
    value: ""

- name: Configurar slapd/password_mismatch
  ansible.builtin.debconf:
    name: slapd
    question: slapd/password_mismatch
    vtype: note
    value: ""

- name: Configurar slapd/postinst_error
  ansible.builtin.debconf:
    name: slapd
    question: slapd/postinst_error
    vtype: note
    value: ""

- name: Configurar slapd/purge_database
  ansible.builtin.debconf:
    name: slapd
    question: slapd/purge_database
    vtype: boolean
    value: false

- name: Configurar slapd/upgrade_slapcat_failure
  ansible.builtin.debconf:
    name: slapd
    question: slapd/upgrade_slapcat_failure
    vtype: error
    value: ""

- name: Instalar paquete slapd
  ansible.builtin.apt:
    name: slapd
    state: present
    update_cache: yes

# slapd shared/organization string nodomain
# slapd slapd/domain string nodomain
# slapd slapd/dump_database select when needed
# slapd slapd/dump_database_destdir string /var/backups/slapd-VERSION
# slapd slapd/internal/adminpw password abc123.
# slapd slapd/internal/generated_adminpw password abc123.
# slapd slapd/invalid_config boolean true
# slapd slapd/move_old_database boolean true
# slapd slapd/no_configuration boolean false
# slapd slapd/password1 password
# slapd slapd/password2 password
# slapd slapd/password_mismatch note
```

```
# slapd slapd/postinst_error note
# slapd slapd/purge_database boolean false
# slapd slapd/upgrade_slapcat_failure error
```

Ejercicios para repasar los conceptos anteriores

Se propone la [Tarea 4](#) para repasar los conceptos vistos hasta este punto.