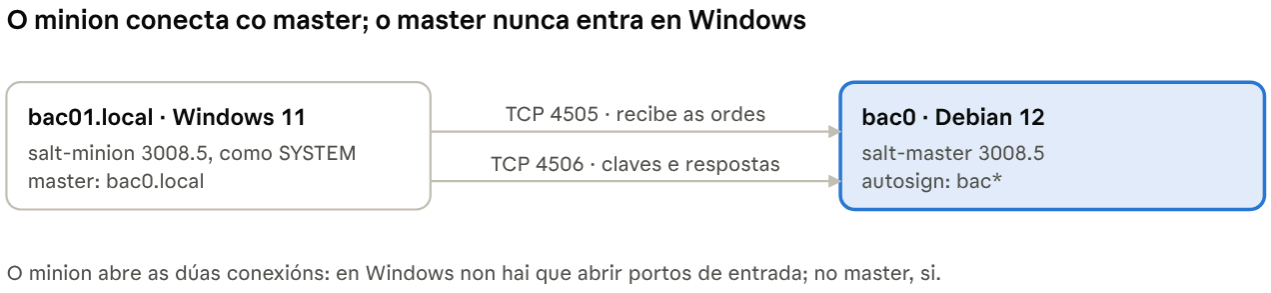
Salt: configuración manual entre Linux (master) e Windows (minion)

Nov 23, 2026 · @Martín Pardiñas

Escenario

O equipo Linux `bac0` (Debian 12, 192.168.0.27) é o **master** de Salt, e o Windows 11 `bac01.local` (192.168.0.24) ten o **minion**. O minion é un servizo que corre como SYSTEM: executa en Windows as ordes do master con permisos de administrador, aínda que a conta que usa o alumnado sexa estándar.



O minion abre as dúas conexións: en Windows non hai que abrir portos de entrada; no master, si.

master e minion · dous portos, conexión iniciada polo minion

A confianza vai por clave: o minion crea un par de claves, o master acepta a súa clave pública e, desde ese momento, só atende a ese minion con esa clave. Os dous equipos usan a mesma versión, 3008.5; o minion nunca debe ser máis novo ca o master.

Parte A. No master Linux (bac0)

A1. Instalar salt-master desde o repositorio oficial

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL
https://packages.broadcom.com/artifactory/api/security/keypair/SaltProjectKey
/public \
  | sudo tee /etc/apt/keyrings/salt-archive-keyring.pgp > /dev/null
curl -fsSL https://github.com/saltstack/salt-install-
guide/releases/latest/download/salt.sources \
  | sudo tee /etc/apt/sources.list.d/salt.sources
sudo apt update
sudo apt install salt-master
salt-master --version           # salt-master 3008.5 (Argon)
```

A2. Configurar o master

A configuración propia vai nun ficheiro aparte dentro de `master.d`, sen tocar o `master` orixinal:

```
sudo tee /etc/salt/master.d/aula.conf > /dev/null <<'EOF'
file_roots:
  base:
    - /opt/piztu/salt
autosign_file: /etc/salt/autosign.conf
EOF
```

- `file_roots`: onde busca o master os estados (`.sls`) e scripts que manda aos minions.
- `autosign_file`: lista de ids de minion que se aceptan sen intervención (paso A3). Se non queres aceptación automática, quita esta liña.

En bac0 este ficheiro chámase `piztu.conf` e xérao Piztu; se usas Piztu, non o edites a man.

A3. (Opcional) Aceptar automaticamente os minions da aula

```
echo 'bac*' | sudo tee /etc/salt/autosign.conf
```

Acepta sen preguntar calquera minion cuxo id empece por `bac`. É cómodo, pero tamén aceptaría un equipo alleo que se presente con ese nome. Sen autosign, cada clave acéptase a man (parte C).

A4. Arrincar o master e abrir os portos

```
sudo systemctl enable --now salt-master
sudo systemctl restart salt-master          # para aplicar a configuración de
A2
sudo ss -tlnp | grep -E ':450[56]'          # debe escoitar en 4505 e 4506
```

Se o master ten cortalumes, abre os dous portos de entrada. Con ufw:

```
sudo ufw allow 4505:4506/tcp
```

En bac0 non hai cortalumes activo, así que este paso non fixo falta.

Anota como chegar ao master desde Windows: o nome `bac0.local` (mDNS) ou a IP `192.168.0.27`. Se o nome `.local` falla nalgún equipo, usa a IP.

Parte B. En Windows (bac01), cunha conta de administrador

A conta estándar do alumnado non pode instalar servizos. Inicia sesión cunha conta de administrador e abre unha **PowerShell como administrador**.

B1. Definir os datos

```
$master = 'bac0.local'                          # ou a IP: 192.168.0.27
$id     = 'bac01.local'                         # nome do equipo tal como
estará no master
$ver    = '3008.5'                              # a mesma versión ca o master
```

O `id` é o nome co que o master coñecerá o equipo. Cómpre que sexa único e, se usas Piztu, igual ao nome do inventario.

B2. Descargar e instalar o minion

```
$msi = "Salt-Minion-$ver-Py3-AMD64.msi"
Invoke-WebRequest -UseBasicParsing -OutFile "$env:TEMP\$msi" `
    -Uri "https://packages.broadcom.com/artifactory/saltproject-
generic/windows/$ver/$msi"
Start-Process msiexec.exe -Wait -ArgumentList "/i `"$env:TEMP\$msi`" /qn
/norestart MASTER=$master MINION_ID=$id START_MINION=1"
Remove-Item "$env:TEMP\$msi"
```

O instalador garda o master e o id no ficheiro `C:\ProgramData\Salt Project\Salt\conf\minion` e deixa o servizo arrincado.

B3. Comprobar a configuración e o servizo

```
Select-String -Path 'C:\ProgramData\Salt Project\Salt\conf\minion' -Pattern
'^(master|id):'
Set-Service salt-minion -StartupType Automatic
Get-Service salt-minion                         # Status: Running
Test-NetConnection $master -Port 4506           # TcpTestSucceeded : True
```

`Test-NetConnection` pode mostrar antes unha `ADVERTENCIA` de que fallou por IPv6 (`fe80::…`). Non importa se ao final di `True`.

Se hai que cambiar o master ou o id despois, mellor nun ficheiro aparte, que non se perde ao actualizar o minion:

```
$conf = 'C:\ProgramData\Salt Project\Salt\conf\minion.d'
Set-Content -Encoding ascii -Path "$conf\aula.conf" -Value "master: $master",
"id: $id"
Restart-Service salt-minion
```

No cortalumes de Windows non hai que abrir nada: o minion só fai conexións de saída cara ao master.

B4. Ver a clave pública do minion

```
Get-Content 'C:\ProgramData\Salt Project\Salt\conf\pki\minion\minion.pub' |
Select-Object -Last 3
```

Anota as dúas liñas anteriores a `-----END PUBLIC KEY-----`: na parte C comprábase que o master ten a mesma clave.

Non compares pegadas. `salt-call --local key.finger` e o hash do ficheiro dan valores distintos da pegada de `salt-key -f`, aínda que a clave sexa a mesma: en bac01 saíron tres pegadas diferentes para unha soa clave. O texto da clave si coincide.

Parte C. Aceptar a clave do minion (no master)

Uns segundos despois de arrincar, o minion envía a súa clave ao master:

```
sudo salt-key -L                          # bac01.local en "Unaccepted Keys"
(ou xa en "Accepted", con autosign)
sudo salt-key -p bac01.local | tail -3    # as últimas liñas teñen que ser as
mesmas do paso B4
sudo salt-key -a bac01.local -y           # aceptala (non fai falta se a
aceptou o autosign)
```

Se non aparece en ningún sitio, o minion non chega ao master: mira a táboa de problemas.

Para comprobar desde que IP chega o minion:

```
sudo timeout 60 tcpdump -nn -i any -c 2 'tcp dst port 4506 and tcp[tcpflags]
& tcp-syn != 0'
# só debe aparecer 192.168.0.24 (bac01)
```

Parte D. Comprobar a comunicación (no master)

```
sudo salt bac01.local test.ping                              # True
sudo salt bac01.local test.version                           # 3008.5
sudo salt bac01.local grains.item id os osrelease            # bac01.local,
Windows, 11
sudo salt bac01.local cmd.run 'whoami'                       # nt
authority\system
```

O último comando mostra que as ordes se executan como SYSTEM, non coa conta do alumnado. Unha orde de exemplo que precisa permisos de administrador:

```
sudo salt bac01.local cmd.run 'netsh advfirewall show allprofiles state'
```

Con isto a comunicación queda configurada. Se usas Piztu, escolle o motor **Salt** para os equipos Windows.

Partir de cero: borrar a configuración previa

Fai os pasos nesta orde. Co autosign activo, se borras a clave no master co minion aínda en marcha, o master volve aceptar ao momento a clave vella.

1. **Windows** (PowerShell como administrador): parar o minion e borrar a súa identidade.

```
Stop-Service salt-minion
$conf = 'C:\ProgramData\Salt Project\Salt\conf'
Remove-Item -Recurse -Force "$conf\pki\minion" -ErrorAction SilentlyContinue
Remove-Item -Force "$conf\minion_id" -ErrorAction SilentlyContinue
```

2. **Master**: borrar a clave do minion.

```
sudo salt-key -d bac01.local -y     # borra a aceptada e a rexeitada
sudo salt-key -L                    # bac01.local xa non aparece
```

3. **Windows**: arrincar o minion, que xera unha clave nova.

```
Start-Service salt-minion
```

4. Segue na parte C.

Para desinstalar o minion por completo: Configuración → Aplicacións → *Salt Minion* → Desinstalar, e despois borra `C:\ProgramData\Salt Project`.

Problemas frecuentes

Rexistros: no master, `/var/log/salt/master`; en Windows, `C:\ProgramData\Salt Project\Salt\var\log\salt\minion`.

Síntoma	Causa	Arranxo
`Minion did not return` e o minion está en Accepted e en Denied	O minion cambiou de clave (reinstalado ou rexenerado)	Comprobar a IP e o texto da clave (B4 e C) e despois `sudo salt-key -d bac01.local -y` para que volva entrar
O rexistro do master repite `the public keys did not match`	O mesmo caso: a clave nova non coincide coa aceptada	O mesmo arranxo
O minion non aparece en `salt-key -L`	Non chega ao master: master mal escrito, `.local` sen resolver ou portos pechados	B3: revisar `master:` e `Test-NetConnection`; probar coa IP; abrir 4505-4506 no master (A4)
Aparece en Unaccepted e non pasa a Accepted	Non hai autosign ou o id non encaixa co patrón	`sudo salt-key -a bac01.local -y`
`Minion did not return` e o equipo está apagado ou noutro sistema	O minion non está correndo	Acender o equipo; en Windows, `Get-Service salt-minion`
`salt-call --local key.finger` dá unha pegada distinta da do master	`salt-call` non calcula a pegada igual ca o master; en bac01 pasou coa clave correcta	Comparar o texto da clave (B4 e C), non as pegadas
Dous equipos co mesmo id	Imaxe clonada ou id repetido	Darlle a cada un un `id` único (B3) e rexenerar a clave (partir de cero)