Acceso SSH sen contrasinal: Linux → Windows 11 con conta estándar

Nov 23, 2026 · @Martín Pardiñas

Escenario

O equipo Linux `bac0` (usuario `usuario`) conecta por SSH ao Windows 11 `bac01.local` sen pedir contrasinal. En Windows entra a conta `usuario`, que é unha **conta estándar** (limitada). Os dous equipos están na mesma rede e o nome `.local` resólvese por mDNS.

Que implica que a conta sexa estándar:

Tarefa	Quen a fai
Instalar e arrincar OpenSSH Server	Un administrador, unha vez (parte A)
Crear a conta e quitala de Administradores	Un administrador, unha vez (parte A)
Copiar a clave pública	A propia conta estándar, por SSH (paso 7)
Cambiar `sshd_config` ou o shell por defecto	Só un administrador

Cunha conta estándar a clave vai no perfil, en `C:\Users\usuario\.ssh\authorized_keys`. O ficheiro `C:\ProgramData\ssh\administrators_authorized_keys` só se usa para contas de administrador.

Os comandos de Linux van en bash; os de Windows, en PowerShell salvo que se indique cmd. Todos os pasos probáronse en bac01 o 2026-11-23 coa conta `usuario` xa estándar, co shell de OpenSSH en PowerShell e tamén en cmd.

Parte A. En Windows, cunha conta de administrador (unha vez)

A conta estándar non pode instalar nin configurar o servidor. Inicia sesión cunha conta de administrador distinta, por exemplo a que se creou ao instalar Windows, e abre unha **PowerShell como administrador**.

```
$u = 'usuario'      # conta estándar que usará o SSH
```

A1. Instalar OpenSSH Server

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
Set-Service -Name sshd -StartupType Automatic
Start-Service sshd
Get-NetFirewallRule -Name OpenSSH-Server-In-TCP     # regra do porto 22,
créase soa
```

A2. Crear a conta estándar, ou quitarlle o permiso de administrador

Se a conta non existe, créaa e métea no grupo Usuarios. Os grupos van polo SID, porque o nome cambia segundo o idioma de Windows.

```
$p = Read-Host "Contrasinal para $u" -AsSecureString
New-LocalUser -Name $u -Password $p -PasswordNeverExpires -
AccountNeverExpires
Add-LocalGroupMember -SID S-1-5-32-545 -Member $u          # Usuarios
```

Se xa existía e era administradora, quítalle ese permiso:

```
Remove-LocalGroupMember -SID S-1-5-32-544 -Member $u -ErrorAction
SilentlyContinue   # Administradores
Get-LocalGroupMember -SID S-1-5-32-544      # $u non debe aparecer
```

O cambio aplícase no seguinte inicio de sesión da conta.

A3. Iniciar sesión unha vez coa conta estándar

Pecha a sesión de administrador, entra unha vez como `usuario` e sae. Así Windows crea o perfil `C:\Users\usuario`, onde irá a clave.

A4. (Opcional) Limpar restos dunha configuración anterior

Se a conta foi administradora antes, a clave puido quedar no ficheiro de administradores. Cunha conta estándar xa non se usa, así que se pode borrar. Isto ten que facelo un administrador:

```
Remove-Item -Force C:\ProgramData\ssh\administrators_authorized_keys -
ErrorAction SilentlyContinue
```

Se hai outras contas de administrador que entran por SSH con clave, non borres o ficheiro: quita só a liña da clave de bac0.

Parte B. En Linux

1. Comprobar a rede

```
ping -c1 bac01.local          # debe resolver o nome e responder
nc -zv bac01.local 22         # debe dicir "open"
```

2. Borrar a configuración anterior (opcional, para partir de cero)

```
ssh-keygen -R bac01.local
rm -f ~/.ssh/known_hosts.old
rm -f ~/.ssh/id_ed25519 ~/.ssh/id_ed25519.pub ~/.ssh/config   # só se queres
rexenerar todo
```

En Windows, a propia conta estándar pode borrar a súa clave antiga:

```
ssh bac01 "cmd /c rmdir /s /q %USERPROFILE%\.ssh"      # pide contrasinal;
vale con calquera shell
```

3. Xerar o par de claves sen frase de paso

```
ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N "" -C "usuario@bac0"
```

4. Crear o alias de conexión

```
cat > ~/.ssh/config <<'EOF'
Host bac01
    HostName bac01.local
    User usuario
    IdentityFile ~/.ssh/id_ed25519
    IdentitiesOnly yes
EOF
chmod 600 ~/.ssh/config
```

5. Gardar a clave de host de Windows

```
ssh-keyscan -t ed25519 bac01.local >> ~/.ssh/known_hosts
ssh-keygen -lf ~/.ssh/known_hosts
```

Para confirmar que é o equipo correcto, compara a pegada `ED25519` coa do propio Windows. A conta estándar non pode ler a clave de host (`Permission denied`), así que se fai cun administrador ou, se xa tes Salt, desde o master:

```
sudo salt bac01.local cmd.run 'ssh-keygen -lf
C:\ProgramData\ssh\ssh_host_ed25519_key.pub'
# 256 SHA256:IAbZe1ub8PIFg9/1eUTRac+oSR7u9lotw+kl6Yejrwo system@bac01
(ED25519)
```

Nunha PowerShell de administrador en Windows o comando é o mesmo: `ssh-keygen -lf C:\ProgramData\ssh\ssh_host_ed25519_key.pub`.

6. Identificar o shell e confirmar que a conta é estándar

Os dous comandos funcionan con calquera shell e piden o contrasinal de Windows.

```
ssh bac01 "whoami /groups | findstr S-1-5-32-544"
```

Non debe aparecer nada. Se aparece `BUILTIN\Administradores`, a conta aínda é de administrador: volve ao paso A2, ou mira o anexo.

```
ssh bac01 "reg query HKLM\SOFTWARE\OpenSSH /v DefaultShell"
```

- Responde `...\powershell.exe`: o shell é PowerShell, usa o paso **7a**.
- Responde `...\cmd.exe`, ou un erro porque o valor non existe: o shell é cmd, o de serie en Windows. Usa o paso **7b**.
- Se non o tes claro, o paso **7c** (sftp) vale con calquera shell.

7. Copiar a clave pública a Windows (pide o contrasinal por última vez)

A clave vai en `C:\Users\usuario\.ssh\authorized_keys`. Executa só unha variante. Todas sobrescriben o ficheiro, así que borran calquera clave anterior.

7a. Shell PowerShell:

```
ssh bac01 "New-Item -Force -ItemType Directory \$env:USERPROFILE\.ssh | Out-
Null; Set-Content -Encoding ascii -Path
\$env:USERPROFILE\.ssh\authorized_keys -Value '$(cat ~/.ssh/id_ed25519.pub)'"
```

7b. Shell cmd:

```
ssh bac01 "if not exist .ssh mkdir .ssh & echo $(cat ~/.ssh/id_ed25519.pub)>
.ssh\authorized_keys"
```

7c. Calquera shell, por sftp (sftp non executa comandos en Windows):

```
sftp bac01 <<'EOF'
-mkdir .ssh
put /home/usuario/.ssh/id_ed25519.pub .ssh/authorized_keys
EOF
```

O `-` diante de `mkdir` fai que non falle se `.ssh` xa existe.

Cunha conta estándar non fai falta `icacls`. O ficheiro créao a propia conta dentro do seu perfil e herda permisos só para ela, SYSTEM e Administradores, que é o que pide `sshd`.

7d. Sen contrasinal, por Salt. Se o Windows xa ten o minion de Salt aceptado polo master, este pode crear o ficheiro como SYSTEM, sen pedir o contrasinal da conta:

```
K=$(cat ~/.ssh/id_ed25519.pub)
sudo salt bac01.local cmd.run "\$d='C:\Users\usuario\.ssh'; New-Item -
ItemType Directory -Force -Path \$d | Out-Null; Set-Content -Encoding ascii -
Path \$d\authorized_keys -Value '$K'; icacls \$d\authorized_keys /setowner
usuario; icacls \$d\authorized_keys /inheritance:r /grant usuario:F /grant
*S-1-5-18:F /grant *S-1-5-32-544:F" shell=powershell
```

- Todo vai nunha soa liña e sen comiñas dobres dentro: o `cmd.run` de Salt en Windows non executa ben scripts de varias liñas.
- Aquí si fai falta o `icacls`: o ficheiro créao SYSTEM, e hai que deixar `usuario` como propietario e quitar permisos de máis.
- Set-Content substitúe o ficheiro enteiro. Para deixar varias claves, por exemplo tamén a de Piztu (paso 9), escríbeas todas: -Value 'clave1','clave2'.

8. Comprobar

```
ssh -o BatchMode=yes bac01 hostname                        # responde "bac01"
sen pedir nada
ssh bac01 "whoami /groups | findstr S-1-5-32-544"       # non debe mostrar
nada
ssh bac01                                                  # sesión
interactiva
```

Con `BatchMode=yes`, se a clave non vale, ssh falla en vez de pedir contrasinal. Así vése claro se funciona.

9. Clave para Piztu

Piztu non usa `id_ed25519`. Con `ssh_key_file: ""` en `/opt/piztu/config.yaml`, busca en `~/.ssh` por esta orde: `id_rsa_piztu*`, `id_rsa` e `id_rsa_*`. Se non hai ningunha, non pode entrar en ningún equipo, aínda que o `ssh` do terminal funcione.

```
grep '^ssh_key_file' /opt/piztu/config.yaml
ls ~/.ssh/id_rsa*                                    # sen resultados: Piztu
non ten clave

# Crear a clave da aula (mesmo tipo que xera Piztu)
ssh-keygen -t rsa -b 4096 -N "" -C "piztu-aula@bac0" -f
~/.ssh/id_rsa_piztu_aula

# Engadila no Windows, sen borrar a que xa hai (Add-Content), por Salt
K=$(cat ~/.ssh/id_rsa_piztu_aula.pub)
sudo salt bac01.local cmd.run "Add-Content -Encoding ascii -Path
C:\Users\usuario\.ssh\authorized_keys -Value '$K'" shell=powershell

# Probar coa mesma clave que usa Piztu
ssh -o BatchMode=yes -o IdentitiesOnly=yes -i ~/.ssh/id_rsa_piztu_aula
usuario@bac01.local hostname
```

Despois **pecha e abre Piztu**: só busca a clave ao arrancar. Nos equipos Linux da aula, repártea con ⚙ Aula → *Crear e enviar clave SSH*. Se a clave xa existe, Piztu non a volve crear: só a envía.

Estado final

Equipo	Ficheiro	Contido
bac0	`~/.ssh/id_ed25519` e `.pub`	Clave persoal para o terminal (permisos 600 na privada)
bac0	`~/.ssh/id_rsa_piztu_aula` e `.pub`	Clave da aula, a que usa Piztu
bac0	`~/.ssh/config`	Alias `bac01`
bac0	`~/.ssh/known_hosts`	Clave de host de bac01
bac01	`C:\Users\usuario\.ssh\authorized_keys`	As dúas claves públicas; só `usuario`, SYSTEM e Administradores teñen permisos

A conta `usuario` de bac01 é estándar: non está en Administradores.

Problemas frecuentes

Para ver o motivo exacto dun rexeitamento, en Linux usa `ssh -v bac01`. Nos rexistros de Windows, un administrador pode velo con `Get-WinEvent -LogName OpenSSH/Operational -MaxEvents 20`.

Síntoma	Causa	Arranxo
`Permission denied (publickey,...)` e o paso 6 mostra `Administradores`	A conta segue sendo administradora, e para ela `sshd` ignora `~\.ssh\authorized_keys`	Quitala de Administradores (A2) e pechar a sesión, ou ver o anexo
`Permission denied (publickey,...)` cunha conta estándar	O ficheiro non está, está en UTF-16 ou ten permisos demasiado abertos	Repetir o paso 7. Ver o contido con `ssh bac01 "type .ssh\authorized_keys"`
`Falta '(' después de 'if'` ou `El operador & está reservado`	O shell é PowerShell e o comando era de cmd	Usar 7a ou 7c
`"New-Item" no se reconoce`	O shell é cmd e o comando era de PowerShell	Usar 7b ou 7c
`Host key verification failed`	Non está gardada a clave de host de bac01	Paso 5
`Connection refused` ou `timed out`	O servizo `sshd` está parado ou o firewall bloquea o porto 22	Paso A1, cun administrador
Acceso denegado ao instalar programas ou cambiar o sistema por SSH	É o esperado nunha conta estándar	Facelo cun administrador ou por Salt
Deixa de entrar despois de quitar a conta de Administradores	A clave só estaba en administrators_authorized_keys, que para unha conta estándar xa non se le	Poñer a clave no perfil da conta: paso 7, ou 7d por Salt se a conta xa non deixa entrar
O ssh do terminal entra, pero Piztu non conecta	Piztu busca só claves id_rsa_piztu*, id_rsa ou id_rsa_*; non usa id_ed25519	Paso 9: crear id_rsa_piztu_aula, autorizala e reabrir Piztu

Síntoma	Causa	Arranxo
ssh-keygen -lf da Permission denied en Windows	A conta estándar non pode ler as claves de host do servidor	Facelo cun administrador ou por Salt (paso 5)

Anexo: se a conta fose de administrador

Se non se pode quitar a conta de Administradores, a clave ten que ir en `C:\ProgramData\ssh\administrators_authorized_keys`, só con permisos para Administradores e SYSTEM. Os SID fan que funcione en calquera idioma de Windows.

Shell PowerShell:

```
ssh bac01 "Set-Content -Encoding ascii -Path
C:\ProgramData\ssh\administrators_authorized_keys -Value '$(cat
~/.ssh/id_ed25519.pub)'; icacls
C:\ProgramData\ssh\administrators_authorized_keys /inheritance:r /grant *S-1-
5-32-544:F /grant *S-1-5-18:F"
```

Shell cmd:

```
ssh bac01 "echo $(cat ~/.ssh/id_ed25519.pub)>
C:\ProgramData\ssh\administrators_authorized_keys & icacls
C:\ProgramData\ssh\administrators_authorized_keys /inheritance:r /grant *S-1-
5-32-544:F /grant *S-1-5-18:F"
```

As dúas variantes probáronse en bac01 e entran sen contrasinal. Para unha conta de administrador sftp non abonda, porque despois fai falta o `icacls`.