mikrotik

- MikroTik — Management VLAN via SSH
- MikroTik — Configurare porte ACCESS e TRUNK (riferimento)
- Guida Accesso Sicuro con Chiavi SSH — MikroTik RouterOS 7.x
- Guida Deploy Template MikroTik — RouterOS 7.x

MikroTik — Management VLAN via SSH

> Comandi testati e funzionanti per gestire un MikroTik via SSH tramite VLAN X di management, che arriva taggata dal trunk su `ether1`.

Prerequisito

Esegui i comandi da una porta da cui **non** ti tagli fuori (porta LAN diversa da ether1, oppure console).

Comandi

```
# 1. Safe Mode
[Ctrl+X]

# 2. Creare il bridge
/interface bridge add name=bridge

# 3. vlanX di management sul bridge
/interface vlan add name=vlanX vlan-id=X interface=bridge
/ip address add address=XXXX/24 interface=vlanX

# 4. ether1 nel bridge
/interface bridge port add bridge=bridge interface=ether1

# 5. tabella VLAN: vlanX taggata su ether1 + alla CPU
/interface bridge vlan add bridge=bridge vlan-ids=X tagged=bridge,ether1
```

```
# 6. firewall SSH
/ip firewall filter add chain=input action=accept protocol=tcp \
  dst-port=22 in-interface=vlanX src-address=XXXX/24 comment="SSH mgmt"

# 7. filtering ON
/interface bridge set bridge vlan-filtering=yes
```

Sposta la regola SSH sopra il drop `!LAN`

```
/ip firewall filter print
/ip firewall filter move [numero-SSH] destination=[numero-drop-!LAN]
```

Test e conferma

Da un dispositivo in VLAN X: `ssh admin@XXXXX`

Se entra, esci dalla Safe Mode con `Ctrl+X` per rendere permanente.

```
/system backup save name=mgmt-ok
/export file=config-mgmt
```

MikroTik — Configurare porte ACCESS e TRUNK (riferimento)

> Guida generica per configurare le porte di un bridge VLAN-aware come **access** (una VLAN, per un PC) o **trunk** (più VLAN taggate, per un apparato VLAN-aware). **Sostituisci i placeholder:**
>
> - `etherX`, `etherY` = porte da configurare
> - `etherUP` = uplink verso lo switch a monte (es. `ether1`)
> - `<VID>` = un ID VLAN (es. `9`, `173`)
> - `<VID_A>`, `<VID_B>` = più ID VLAN Presuppone bridge in `vlan-filtering=yes`.

---

1. Access vs Trunk — il concetto

	ACCESS	TRUNK
VLAN sulla porta	**una sola**	**più di una**
Tag	**untagged** (senza tag)	**tagged** (col tag)
Cosa colleghi	un PC / dispositivo normale	switch, router, AP (VLAN-aware)
PVID	= la VLAN della porta	1 (neutro)

Le tre primitive:

- **tagged** = la VLAN esce/entra col tag → trunk
- **untagged** = la VLAN esce/entra senza tag → access
- **PVID** = a quale VLAN assegnare il traffico untagged in ingresso

>

> ⚠ **Sicurezza:** se metti una porta access sulla **VLAN di management**, chi si collega lì raggiunge SSH/WinBox del router. Comodo come "porta di rientro" fidata, ma non esporla a utenti non affidabili.

---

2. Porta ACCESS

Un PC su `etherX` finisce nella VLAN `<VID>` senza saperlo.

Creare (VLAN nuova)

```
/interface bridge vlan add bridge=bridge vlan-ids=<VID> tagged=etherUP untagged=etherX
/interface bridge port set [find interface=etherX] pvid=<VID>
```

- `tagged=etherUP` → la VLAN arriva taggata dallo switch a monte.
- `untagged=etherX` → esce pulita verso il PC.
- `pvid=<VID>` → il traffico del PC entra nella VLAN giusta.

Aggiungere una porta access a una VLAN che ESISTE già

La VLAN c'è già in tabella → **modifica** (non `add`), aggiungendo la porta agli untagged:

```
/interface bridge vlan set [find vlan-ids=<VID>] tagged=etherUP untagged=etherX
/interface bridge port set [find interface=etherX] pvid=<VID>
```

> `set` riscrive l'intera lista: rielenca i tagged/untagged già presenti che devono restare.

Convertire una porta da TRUNK ad ACCESS

Se la porta era tagged e la vuoi untagged:

```
/interface bridge vlan set [find vlan-ids=<VID>] tagged=etherUP untagged=etherX
/interface bridge port set [find interface=etherX] pvid=<VID>
```

(togli `etherX` dai tagged rielencando solo le porte da mantenere, e mettila negli untagged)

---

3. Porta TRUNK

Su `etherX` colleghi un apparato VLAN-aware. Più VLAN taggate.

Creare

```
# aggiungere la porta al bridge
/interface bridge port add bridge=bridge interface=etherX

# una VLAN
/interface bridge vlan add bridge=bridge vlan-ids=<VID> tagged=etherUP,etherX

# più VLAN nella stessa riga
/interface bridge vlan add bridge=bridge vlan-ids=<VID_A>,<VID_B> tagged=etherUP,etherX

# PVID neutro (di solito già 1)
/interface bridge port set [find interface=etherX] pvid=1
```

Aggiungere una porta a una VLAN già esistente

```
/interface bridge vlan set [find vlan-ids=<VID>] tagged=etherUP,etherX
```

> ⚠ `set tagged=...` **sovrascrive**: rielenca tutte le porte da mantenere.

Native VLAN (una VLAN untagged sul trunk)

```
/interface bridge vlan set [find vlan-ids=<VID>] tagged=etherUP untagged=etherX
/interface bridge port set [find interface=etherX] pvid=<VID>
```

---

4. Togliere / modificare

```
# togliere una VLAN SOLO da una porta (riscrivi senza quella porta)
/interface bridge vlan set [find vlan-ids=<VID>] tagged=etherUP

# rimuovere una VLAN da TUTTE le porte
/interface bridge vlan remove [find vlan-ids=<VID>]

# togliere la porta dal bridge del tutto
/interface bridge port remove [find interface=etherX]

# cambiare il PVID
/interface bridge port set [find interface=etherX] pvid=1
```

Disabilitare (granularità)

Obiettivo	Comando
VLAN off su **tutto lo switch**	`/interface bridge vlan disable [find vlan-ids=<VID>]`
VLAN off **solo su una porta**	non c'è `disable` per-porta: usa `set` togliendo la porta

---

5. Verifica

```
/interface bridge vlan print detail                 # config reale (tagged/untagged)
/interface bridge port print                        # PVID, flag hw-offload (H), inactive
(I)
```

```
/interface bridge host print where interface=etherX    # MAC appresi sulla porta
/interface bridge host print where interface=etherX vid=<VID>
```

> **`tagged` vs `current-tagged`:** `tagged` = configurato (sempre); `current-tagged` = attivo ora (solo porte up). Porta scollegata (`I`) → non appare in `current-tagged` ma la config c'è: guarda `print detail`.
>
> **Filtri host:** i campi sono `interface` e `vid`, NON `on-interface`/`vlan-id`. Dopo `where` premi **Tab** per l'elenco dei campi validi.

---

6. Lato switch a monte (Cisco)

Ogni VLAN usata sul MikroTik deve essere **allowed** anche sul trunk dell'apparato a monte:

```
vlan <VID_A>
vlan <VID_B>
interface <porta-verso-mikrotik>
 switchport mode trunk
 switchport trunk allowed vlan add <VID_A>,<VID_B>
```

---

7. Trappole comuni (imparate sul campo)

1. **`set tagged=...` / `untagged=...` SOVRASCRIVE** la lista intera — rielenca sempre le porte da mantenere, o le perdi.
2. **`add` di una VLAN che esiste già → `failure: vlan already added`**, e fallisce l'INTERO comando (anche le altre VLAN della stessa riga). Se una VLAN esiste (es. la management), aggiungi la porta con `set`, non con `add`.
3. **Access = untagged + PVID.** Servono entrambi: senza PVID il traffico in ingresso dal PC finisce in VLAN 1.
4. **Trunk = tagged + PVID 1.** Il PVID resta neutro perché tutto viaggia taggato.
5. **VLAN di management:** modificandola, mantieni sempre `tagged=bridge,etherUP` o perdi l'accesso. Lavora in **Safe Mode** (`Ctrl+X`).

6. **Config su entrambi i lati:** la VLAN va permessa sia sul MikroTik sia sull'apparato a monte.
7. **Porta scollegata (`I`):** verifica con `print detail`, non col riepilogo.

---

8. Chiusura (quando funziona)

```
[Ctrl+X]                                  # esci da Safe Mode = rende permanente
/system backup save name=porte-vlan-ok
/export file=config-porte-vlan
```

Scarica i file dal menu **Files** e conservali fuori dal router.

Guida Accesso Sicuro con Chiavi SSH — MikroTik RouterOS 7.x

Configurazione dell'autenticazione a chiave pubblica su RouterOS: più sicura delle password e ideale per automatizzare SCP/SSH verso più router da remoto.

> **Regola d'oro:** non disabilitare mai l'accesso via password PRIMA di aver verificato al 100% che l'accesso con chiave funziona. Altrimenti rischi il lock-out, soprattutto da remoto.

---

1. Genera la coppia di chiavi (sul tuo PC)

```
ssh-keygen -t ed25519 -f ~/.ssh/mikrotik_key
```

- `ed25519` è l'algoritmo moderno consigliato (chiavi corte e robuste).
- Se hai router più vecchi che non lo supportano, usa RSA a 2048+ bit:

  ```
  ssh-keygen -t rsa -b 4096 -f ~/.ssh/mikrotik_key
  ```

- Alla richiesta, imposta una **passphrase**: protegge la chiave privata se qualcuno mette le mani sul file.

Vengono creati due file:

- `mikrotik_key` → chiave **privata** (resta sul tuo PC, NON condividerla)
- `mikrotik_key.pub` → chiave **pubblica** (da caricare sul router)

---

2. Carica la chiave pubblica sul router (via SCP)

```
scp ~/.ssh/mikrotik_key.pub admin@<IP-router>:
```

Se RouterOS 7.x rifiuta la connessione per mismatch di algoritmi, abilita lo strong crypto sul router e riprova (vedi anche passo 5):

```
/ip ssh set strong-crypto=yes
```

Verifica che il file sia arrivato:

```
/file print
```

Deve comparire `mikrotik_key.pub`.

---

3. Importa la chiave su RouterOS

Collegati via SSH al router e associa la chiave pubblica all'utente:

```
/user ssh-keys import public-key-file=mikrotik_key.pub user=admin
```

Verifica che sia stata importata:

```
/user ssh-keys print
```

---

4. Verifica l'accesso con chiave (dal tuo PC)

```
ssh -i ~/.ssh/mikrotik_key admin@<IP-router>
```

Se entri **senza** che il router chieda la password (al massimo ti viene chiesta la passphrase della chiave), l'autenticazione a chiave funziona.

> **Non procedere al passo 5 finché questo test non riesce in modo affidabile.**

---

5. Rafforza la sicurezza (solo DOPO aver validato il passo 4)

5.1 Strong crypto

Forza algoritmi robusti e disabilita quelli deboli/obsoleti:

```
/ip ssh set strong-crypto=yes
```

5.2 Disabilita il login con password

Da qui in poi si entra SOLO con chiave:

```
/ip ssh set password-authentication=no
```

> **ATTENZIONE:** dopo questo comando, chi non ha la chiave non entra più via SSH. Assicurati che:
>
> - la tua chiave privata sia al sicuro e con un backup;
> - resti almeno un canale di recupero (vedi sezione 6).

---

6. Rete di sicurezza (paracadute)

Con la password SSH disabilitata, se perdi la chiave privata o cambi PC senza aver configurato la nuova chiave, non entri più via SSH. Tieni quindi attivi:

- **MAC-Winbox su tutte le interfacce** come accesso di emergenza a Layer 2:

```
/tool mac-server set allowed-interface-list=all
/tool mac-server mac-winbox set allowed-interface-list=all
```

- Un **backup della chiave privata** in luogo sicuro.
- Idealmente, accesso fisico o una persona sul posto per i casi estremi.

Config lato client (comodità)

Nel tuo `~/.ssh/config` puoi definire scorciatoie per non ripetere `-i` ogni volta:

```
Host mikrotik-sedeX
    HostName 172.17.5.116
    User admin
    IdentityFile ~/.ssh/mikrotik_key
```

Poi accedi semplicemente con:

```
ssh mikrotik-sedeX
```

Checklist finale

- ☐ Coppia di chiavi generata con passphrase
- ☐ Chiave pubblica caricata e importata sul router
- ☐ Accesso con chiave testato e funzionante
- ☐ Strong crypto abilitato
- ☐ Login password disabilitato SOLO dopo il test riuscito
- ☐ Paracadute attivo (MAC-Winbox + backup chiave privata)

Guida Deploy Template MikroTik — RouterOS 7.x

Procedura per applicare da remoto un template di configurazione base (accesso di management via VLANX + SSH + paracadute MAC) su router MikroTik, in modo pulito e ripetibile.

> **Principio di fondo:** il template applica solo il MINIMO per rientrare. Le VLAN dati e le regole di dettaglio si aggiungono dopo, da remoto, una volta ripreso l'accesso. Il router di prova deve avere la STESSA versione RouterOS dei router di produzione: un collaudo su versione diversa non è valido.

---

1. Controllo versione e (se serve) aggiornamento

1.1 Verifica la versione attuale

```
/system resource print
```

Guarda il campo `version` (es. `7.21.4`) e `architecture-name`.

Controlla anche il firmware del RouterBoard:

```
/system routerboard print
```

Confronta `current-firmware` con `upgrade-firmware`: se differiscono, andrà aggiornato (passo 1.5).

1.2 Decidi la versione target

Il template va testato e applicato sulla **stessa major/minor version** dei router di produzione. Se la produzione è su una versione precisa (es. 7.21.4), allinea tutti i router a quella, non "all'ultima disponibile".

1.3 Aggiornamento ONLINE (se il router naviga)

Richiede connettività Internet **e DNS funzionante** sul router. Verifica prima:

```
/ping 8.8.8.8
/ping download.mikrotik.com
```

Se il primo va ma il secondo no, manca il DNS:

```
/ip dns set servers=9.9.9.9,149.112.112.112
```

Poi aggiorna:

```
/system package update set channel=stable
/system package update check-for-updates
/system package update download
/system package update install
```

Il comando `install` riavvia il router e applica la nuova versione.

> Per massima stabilità si può usare `channel=long-term` invece di `stable`.

1.4 Aggiornamento OFFLINE (metodo consigliato: non dipende da Internet)

1. Da un PC, scarica da https://mikrotik.com/download il **Main package** per architettura **ARM** della versione desiderata (file `.npk`).
2. Carica il pacchetto sul router via SCP:
   ```
   scp routeros-7.xx.x-arm.npk admin@<IP-router>:
   ```
3. Verifica che sia arrivato:

```
/file print
```

4. Riavvia: RouterOS installa il pacchetto all'avvio.

```
/system reboot
```

5. A riavvio completato, verifica la nuova versione:

```
/system resource print
```

1.5 Aggiorna il firmware del RouterBoard (dopo l'update di RouterOS)

```
/system routerboard print
```

Se `current-firmware` != `upgrade-firmware`:

```
/system routerboard upgrade
/system reboot
```

1.6 Backup prima di operare (sempre consigliato)

```
/system backup save name=pre-update
/export file=config-pre-update
```

---

2. Upload del template ed esecuzione

2.2 Carica il template via SCP

Dal tuo PC:

```
scp template-base-mgmt.rsc admin@<IP-router>:
```

Se RouterOS 7.x rifiuta la connessione per mismatch di algoritmi SSH, abilita lo strong crypto sul router e riprova:

```
/ip ssh set strong-crypto=yes
```

2.3 Verifica che il file sia arrivato col nome ESATTO

Collegati via SSH al router e:

```
/file print
```

Deve comparire `template-base-mgmt.rsc`. **Il nome deve combaciare esattamente** con quello usato nel comando del passo 2.4: se non combacia, il reset azzera il router ma NON esegue lo script.

2.4 Esegui il template con reset-con-script

```
/system reset-configuration no-defaults=yes skip-backup=yes run-after-reset=template-base-mgmt.rsc
```

Il router si azzera, applica il template e si riavvia.

2.5 Riconnetti

Dopo ~1–2 minuti, riprova l'accesso all'IP di management:

```
ssh admin@<IP-management-sede>
```

Se risponde, il deploy è riuscito. Da qui completi il resto della configurazione da remoto (VLAN dati, firewall, ecc.).

Paracadute

Il template abilita **MAC-Winbox su tutte le interfacce**: se la VLAN/trunk non passa, chiunque sul posto con un cavo e Winbox può entrare via MAC. Il `run-after-reset` **non ha rollback**: una volta

azzerato, o il template funziona o serve un intervento fisico. Per questo il collaudo in laboratorio sulla stessa versione deve essere sempre verificato al 100%.

---

3. Il template: spiegazione e manutenzione

3.1 Perché NON usare le variabili globali nel template

Una versione precedente usava `:global` e `$MGMTIP` in cima al file. In un file `.rsc` **importato**, l'espansione delle variabili è fragile e cambia tra versioni di RouterOS: produce errori di parsing tipo `expected end of command` o `expected interface value`. La versione con **valori diretti** è robusta e portabile tra versioni. Regola: nei file di configurazione da importare, usa valori diretti, non variabili.

3.2 Mantenere il template per versioni future di RouterOS

La sintassi di alcuni comandi (soprattutto **bridge VLAN filtering**, routing) cambia tra major version (es. v6 → v7). Per ogni nuova versione RouterOS che adotterai:

1. **Aggiorna un router di prova** alla nuova versione (stessa dei futuri router di produzione).
2. **Reimporta il template a mano** e verifica che non dia errori:

   ```
   /import file-name=template-base-mgmt.rsc
   ```

   Se una riga fallisce, l'import indica riga e colonna: correggi la sintassi.

3. **Rigenera il template dal router aggiornato** partendo da una config buona:

   ```
   /export file=nuovo-template
   ```

   e ripulisci l'export tenendo solo le righe essenziali (bridge, VLAN mgmt, IP, route, DNS, mac-server), reinserendo i marcatori `### MODIFICA`.
4. **Valida il run-after-reset** sul banco prima di usarlo in produzione.

5. Tieni un template **per major version** se gestisci un parco misto (es. `template-v6.rsc`, `template-v7.rsc`), perché la stessa sintassi non è garantita compatibile.

3.3 Checklist di collaudo prima di ogni uso in produzione

- ☐ Router di prova sulla STESSA versione dei router di produzione
- ☐ Import manuale del template SENZA errori
- ☐ Raggiungibilità via SSH ATTRAVERSO IL TRUNK (switch a monte come in sede, VLANX taggata, native allineata)
- ☐ `run-after-reset` riporta il router su in modo affidabile e ripetibile
- ☐ MAC-Winbox raggiungibile come paracadute