Dal palco al cloud:
la nostra WebRadio
con Linux e Open Source

Associazione DN Service - DN Music APS

Relatori:

Domenico De Luca e Noè Mollica

Introduzione

- Profilo Instagram _dn.music e progetto 'Dream Catchers'
- Obiettivo: diffondere la cultura musicale e creare uno spazio per i giovani
- Sede DN Lab – Avellino, via Vicolo Posillipo 21
- Sala insonorizzata e attrezzata per registrare e trasmettere tramite WebRadio

Obiettivo del Talk

- La progettazione e realizzazione di una WebRadio completamente basata su software open source, garantendo sicurezza, affidabilità e accesso multi-sede.

Architettura e Flusso Dati

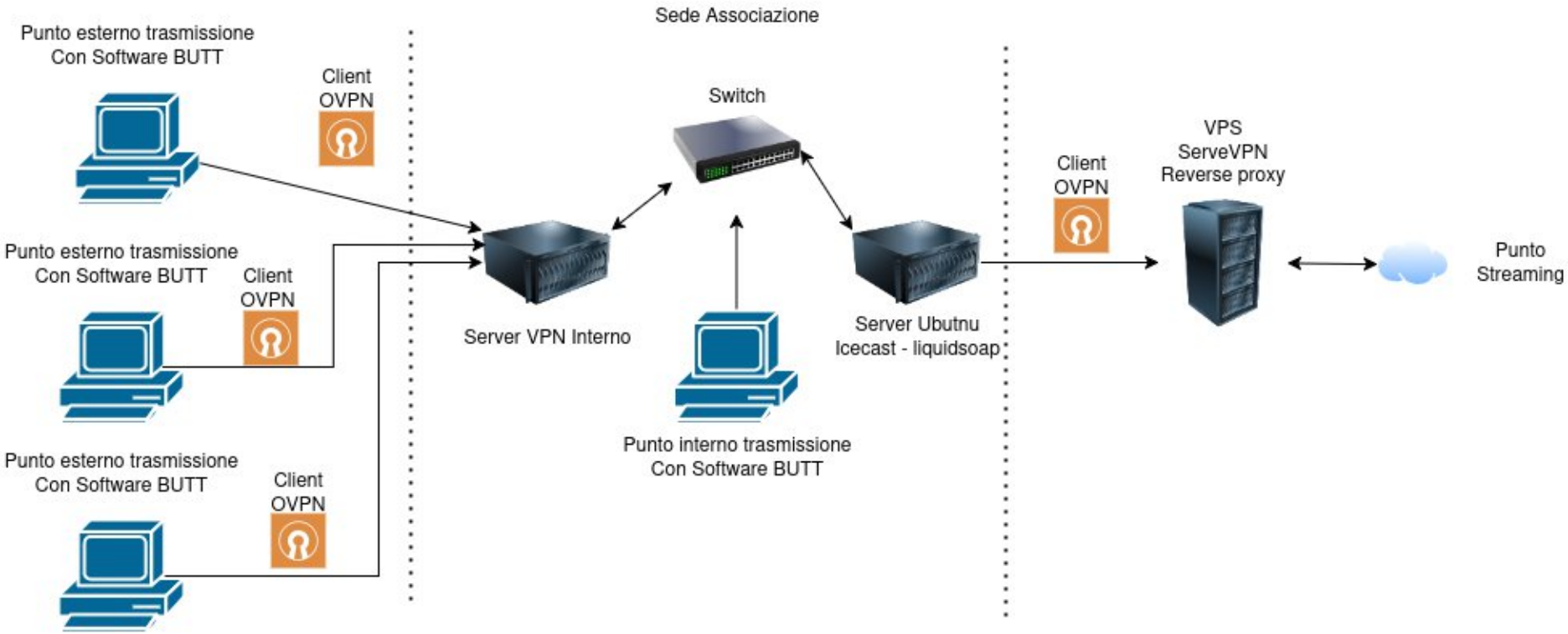
• Sede DN Lab: server Ubuntu (Icecast + Liquidsoap), server OpenVPN interna

• Postazioni esterne: PC con BUTT interne o esterne via OpenVPN

• VPS esterno: reverse proxy pubblico con Nginx e server OpenVPN

• Icecast non esposto direttamente, flusso tramite VPN e proxy sicuro

Architettura e Flusso Dati



Sicurezza e Isolamento dei Servizi

- Nessun accesso diretto a Icecast o Liquidsoap
- VPN dedicate per DJ e per connessione al VPS
- Certificati personalizzati /32 e revocabili singolarmente:

Es. push “route 192.168.1.120 255.255.255.255”

- Doppia VPN = massima protezione e isolamento
- Tutto il traffico avviene in canali cifrati

Configurazione Icecast (estratto)

```
<limits>
 <clients>100</clients>
 <sources>10</sources>
</limits>
<mount-name>/live</mount-name>
<password>live</password>
```

Configurazione Liquidsoap (estratto)

- Sorgenti HTTP multipli per DJ remoti
- Fallback automatico verso blank() senza interrompere il flusso
- Mix tra più DJ e playlist di backup
- Fallback aggiornamento automatico mediante una playlist di file /srv/radio/*.mp3

Sorgenti HTTP dei due DJ, entrambe

dj1 = input.http("http://localhost:8000/stream1")
dj2 = input.http("http://localhost:8000/stream2")

Entrambe queste sorgenti sono considerate **“fallibili”**

ogni flusso viene “wrappato” (incapsulato) all’interno di un **meccanismo di fallback**:

safe_dj1 = fallback(track_sensitive=false, [dj1, blank()])
safe_dj2 = fallback(track_sensitive=false, [dj2, blank()])

Playlist locale di backup

Per garantire che, in caso di indisponibilità totale dei DJ, la radio continui a trasmettere, viene predisposta una playlist locale:

pl = playlist("/srv/radio/.mp3", reload_mode="watch")*
safe_playlist = fallback(track_sensitive=false, [pl, blank()])

Fallback principale: prima DJ mix, poi playlist
Il flusso principale della radio viene definito con un fallback gerarchico:

radio = fallback(track_sensitive=false, [mixed, safe_playlist])

La logica è la seguente:

- Priorità 1 → il mix dei due DJ (mixed)
- Priorità 2 → la playlist locale (safe_playlist)
- Se nessuna delle due è disponibile, si ricade implicitamente sul silenzio

Configurazione Nginx Reverse Proxy (estratto)

- Ascolto su 443 SSL – Certificati TLS 1.2/1.3
- Suite di cifratura moderne (AES-GCM)
- proxy_pass http://10.10.0.3:8000/live
- Header: Host, X-Real-IP
- Icecast non esposto pubblicamente

Conclusioni

- Webradio sicura, scalabile e open source
- Doppia VPN e isolamento garantiscono continuità operativa
- Architettura replicabile per altre realtà associative o educative