Nguyen Vu Minh Tuan

VoIP Engineer | Telecom Infrastructure | AI Automation

Ho Chi Minh City, Vietnam

Email LinkedIn GitHub

About Me

Hi, I'm Tuan — a VoIP engineer, telecom infrastructure builder, and AI automation enthusiast. Over 6+ years I've gone from building PBX systems and SBC solutions to operating one of the largest FreeSWITCH-based voice platforms in the United States — from SIP protocol debugging and C/C++ systems programming to CI/CD pipelines and production fleet management.

At Telnyx, I run the FreeSWITCH-based voice platform that powers millions of calls a day across 18 datacenters. I own the full lifecycle: shipping 200+ releases through phased rollouts, building the CI/CD pipeline from code merge all the way to fleet deployment, setting up 800+ alert rules, and serving as on-call for a fleet handling 100,000+ concurrent channels.

What sets me apart is that I don't just fix problems — I automate them away. I built an AI operations platform purpose-built for FreeSWITCH that handles crash triage, anomaly detection, and deduplication around the clock. It's changed how I work — the routine is handled by AI, so I can focus on the hard problems that need a human.

Every incident becomes a pattern, every pattern becomes automation. That's the mindset I bring to any team — leave the system better than I found it.

I once heard: "Musicians don't retire — they stop when there's no more music in them." I still have a lot of music to offer.

Work Experience

Present — May 2023

TELNYX LLC

FreeSWITCH Operations Engineer

Responsibilities

- Own end-to-end operations of the FreeSWITCH-based B2BUA service — spanning QA, deployment, production operations, troubleshooting, maintenance and decommission
  - Managing 200+ servers and 500+ containers across 18 global datacenters
  - Managed fleet-wide configuration as Infrastructure as Code via Ansible, driving thousands of configuration changes across the fleet
  - Managed full datacenter lifecycle: decommissioned 4 production sites (60+ servers migrated with live traffic) and commissioned 3+ new datacenters including Kubernetes-hosted environments
  - Authored traffic routing PRs managing fleet-wide canary fraction control, per-instance traffic assignment, and version rollouts as code with pre/post live traffic validation via Traffic Director
  - Contributed C/C++ PRs to the enterprise FreeSWITCH fork (FreeSWITCH and FreeSWITCH modules)
- Support B2BUA integration with upstream platform services, interfacing with routing, media, billing, and QA teams
  - Submitted configuration PRs tuning jitter buffer parameters, DTMF interop settings, and multi-sample codec configurations across the fleet
  - Managed customer-specific routing overrides for enterprise accounts with varying channel capacities and dedicated server isolation
  - Collaborated across platform, media, routing, and QA teams to drive incident resolution and contribute to ad-hoc operational needs
- Diagnose and resolve latency, scalability, and performance issues across the distributed fleet
  - Optimized jitter buffer configuration, codec negotiation, and SRTP key rotation for fleet-wide performance
  - Operated centralized log analysis infrastructure (Graylog, Grafana/Loki) for fleet-wide search and cross-datacenter correlation
  - Cross-analyzed calls using application logs, CDRs, dialplan traces, SIP captures, and PCAP/RTP flows to isolate root causes across distributed service layers

    - Validated performance fixes using controlled SIPp load test scenarios at 3k–50k calls per instance before production rollout
    - Applied cross-repository debugging methodology, tracing issues through 4+ repositories (e.g., consul-kv debug configs → freeswitch core fix → b2bua version bump → tel-core-alerts alerting) to isolate single root causes
- Handle cross-team and customer escalations for B2BUA-related issues
    - Traced complex multi-hop call flows (5–7 legs) across distributed systems involving transfers, conferences, and multiple service instances
    - Reproduced customer-reported issues using a softphone + SIPp + n8n + custom scripts + webhook reproduction environment with testing different codec modes and frame alignment combinations
    - Captured and analyzed PCAPs at every call leg (ingress, internal, egress) to isolate root cause across service boundaries
- Build and maintain monitoring and alerting infrastructure to ensure production system health
    - Built and maintained Prometheus alert rules across 18 datacenter-specific configurations with P0–P3 severity-based incident routing
    - Designed custom PromQL alert patterns: quantile-based anomaly detection, traffic routing correlation, API call saturation, packet loss, login rate detection, and per-anchorsite ASR/MOS anomaly detection
    - Implemented maintenance-aware alert suppression via service health status, preventing false positives during scheduled maintenance windows
- Deploy releases and hot-fixes via zero-downtime phased deployment strategy
    - Delivered B2BUA releases, maintaining weekly-to-bi-weekly cadence for a service handling millions of daily calls
    - Executed gradual deployment with canary validation at each stage, including revert-and-redeploy cycles when regressions detected
    - Coordinated with QA, development, and operations teams across release cycles to ensure deployment safety
    - Monitored deployments in real time and stayed ready to roll back during every release window
    - Managed Debian and Ubuntu version upgrade for B2BUA service
- Automate deployment pipelines and operational processes to accelerate release cycles and capacity scaling
    - Created a centralized deployment repository with automated chain: code merge → package build → container image → dev fleet deployment
    - Integrated health-check and log-check monitoring into deployment windows for automated regression detection during rollouts
- Serve in on-call rotation and perform ongoing maintenance of the B2BUA service

- Served as DRI coordinating multi-incident resolution across the global fleet
- Designed and executed emergency rollback procedures across 200+ servers spanning 18 datacenters under time pressure
- Responded to P0–P3 incidents within defined SLA response times, escalating cross-service issues to appropriate squads
- Set up weekly crash reporting with pattern grouping, cross-ticket mapping, and deduplication across dual alert channels
- Execute emergency traffic shifts during incident response to redirect live call traffic away from affected servers or datacenters

Initiative

- Designed and built Telnyx Internal AI workflow for telephony
  - A company-internal automated AI diagnostic pipeline built within Telnyx for internal use, integrated into the Prometheus alert system
  - Achieved 99.5% error log noise reduction (high-volume ERROR logs reclassified to WARNING) for better AI detection
  - 4 workflow types routed to AI triage: process restarts (crash analysis with GDB stack trace reconstruction), per-site answer-seizure-ratio drops, upstream service timeout detection, and MRCP timeout handling
  - YAML-based diagnosis prompts → centralized log search → stack trace analysis → ticket dedup → auto-create tickets; runs fully autonomously 24/7 with analysis posted to team channel
  - Built Grafana/Loki dashboards covering fleet health, audio quality, BGAPI thread pool utilization, and per-datacenter deployment tracking across 20+ metric families
  - Replaced traditional alert resolution runbooks with Telnyx Internal AI diagnostic workflows
- Led Jenkins → GitHub Actions migration across the core telephony platform
  - Implemented tag-release workflows replacing legacy post-commit hooks with multi-branch build support, secrets management, and vulnerability scanning
  - Built cross-repo build trigger infrastructure: code merge → package build → container image → dev fleet deployment
  - Designed B2BUA deployment pipeline with staged execution, parallel builds, automated traffic shifting, manual approval gates, and chat-integrated approval workflows
  - Built post-maintenance watchdog monitoring for automated recovery verification after maintenance windows
- Led Graylog to Grafana/Loki migration for improved log correlation and unified observability
- Created a Python-based out-of-band tester for WebRTC service

- Sends JSON-RPC WebSocket login commands sequentially to every WebRTC instance across all datacenters, validating each instance is responsive and functional
- Captures 4 response states per instance: login success, login failure (credentials issue), timeout (unreachable IP/port or lockup process), and connection refused (port open but JSON-RPC rejected)
- Exposes Prometheus metrics per response state per instance, integrated with Grafana dashboards and alerting rules for automated incident detection
- Deployed as a single container with CI/CD auto-deployment from main branch, using Telnyx SIP credentials stored in Vault

- Designed black-box and load test scenarios for call flows including 3PCC, UPDATE handling, codec renegotiation, and SRTP crypto changes
- Evaluated and built audio quality analysis tooling for automated voice quality assessment in customer escalation workflows
  - Sevana PVQA (Passive Voice Quality Analysis): deployed dockerized service on dev hosts, fed production PCAPs with known good/bad audio, assessed detection accuracy and JSON output integration feasibility
  - Google VisQOL (Virtual Speech Quality Objective Listener): deployed dockerized service on dev hosts, tested MOS-LQO scoring with good/bad B2BUA audio samples
  - Facebook Audiobox Aesthetics: deployed dockerized service on dev hosts, tested with production audio samples, assessed Production Quality and Content Usefulness metrics
  - TestDevLab RTP-Audio-Stat: tested NISQA, SpeechMOS, and barko metrics against production PCAPs, documented results in evaluation spreadsheet
  - Compared tooling approaches for automated audio quality scoring across customer escalation workflows
- Implemented FreeSWITCH unit testing and regression test suites in CI/CD pipeline with custom dialplan entries to trigger specific call behaviors
  - 3PCC, UPDATE handling, codec renegotiation, SRTP crypto changes, and call recovery validation
- Designed and built Council of Claws — an autonomous AI operations platform
  - Built from scratch on dedicated infrastructure (Clawdbot/OpenClaw platform), a production 24/7 autonomous AI operations platform with 6 specialized LLM agents covering crash triage, fleet monitoring, code review, and incident response
  - Automated weekly crash reports with pattern grouping, ticket links, and trend metrics posted to team channel
  - AI-generated PRs merged into production repositories; automated code review assistance integrated into GitHub workflows

- Semantic memory system using PostgreSQL + pgvector with 2,400+ extracted facts, Lossless Context Management (LCM), and daily skill extraction from agent memory logs
- Inbox automation: scans Slack, GitHub notifications, issue tracker, and email every 5 minutes with AI classification and autonomous response
- Self-improving feedback loop: every failure logged, corrections auto-promoted to shared patterns — so we never repeat the same mistake
- Runs on 50+ shared protocols including mandatory peer review, agent accountability shifts, and escalation paths

Incident Response

- Investigated crash cases across 14+ distinct failure families using GDB backtraces, core dumps, and production traffic pattern reproduction
  - Investigated a mass crash event across dozens of servers — analyzed unique crash signatures, identified the majority caused by a system-level file descriptor race condition, and consolidated findings into root cause tickets
  - Established weekly crash quantification cadence: pattern grouping, cross-ticket deduplication, and automated triage transforming raw crash noise into actionable root cause tickets
  - Consolidated multiple simultaneous server crash alerts into root causes covering most of incidents through systematic pattern grouping and cross-ticket deduplication
  - Reproduced crash conditions in isolated test instances using captured production traffic patterns for debugging
- Resolved customer escalations with detailed RCA reports, PCAP analysis, and resolution timelines
  - Investigated billing CDR loss caused by malformed Call-ID headers; resolved garbled audio on long-duration calls via SRTP crypto tag rotation
  - Diagnosed MS Teams DTMF interoperability failures (RFC 2833 violations) and built reusable diagnostic patterns for downstream cases
  - Performed SRTP decryption and encrypted media debugging, including investigation of unauthorized RTP injection across multi-hop call flows spanning 5–7 legs
  - Carrier SIP compatibility fixes — upstream provider peering, header normalization, and routing compatibility issues
- Authored incident response procedures and post-mortems with regression test scenarios covering core protocol failure modes
  - Short-term: created targeted alerts with higher retention to detect recurrence and improve early warning coverage
  - Long-term: built regression test scenarios integrated into deployment pipeline to catch regressions before production rollout

May 2023 — Nov 2021

TEL4VN ICT Training Center

System Team Leader

Responsible:

- Fail-over solution for high availability for FreeSWITCH ▶ Video
  - Using corosync and pacemaker to run FreeSWITCH on a Virtual IP, when failure on the active FreeSWITCH happened, corosync and pacemaker will switch Virtual IP to standby FreeSWITCH server to minimize downtime, in some cases, FreeSWITCH can recover calls on the active server which failed
- Load-balance solution for scaling concurrent calls on FreeSWITCH ▶ Video
  - Coding OpenSIPs with module mid_registrar, load_balancer, and dispatcher to load-balancing SIP traffic with round robin or weighted order on 3 FreeSWITCH servers to reach 5,000 extensions and 1,200 concurrent calls
  - Load test with SIPP makes calls to this OpenSIPs - 3 FreeSWITCH server for multiple days
  - Documented all information about how the system works, load test historic data, how to read a log file, how to deploy
- Use ISO Debian live image to create a custom ISO image containing VoIP System which is documented to install completely offline without a connection to the internet
- Write 2-month plan and personally train interns fresh out of college, or university on SIP, RTP, VoIP Systems, and Linux so they can support all current customers and handle any running project, 3 people have become full-time employees with my guidance
- R&D new VoIP solution
  - Successfully using FreeSWITCH to call video on 2 endpoints using H264 protocol, also enable recordings of both video and audio with mod_av
- Successfully using FreeSWITCH to detect both voice and keypress in the same calls with mod_umimrcp on the FreeSWITCH server to detect voice and keypress to detect medical records for a foreign medical company
  - Write multiple lua files to detect 2 languages English and Spanish, multiple string voice or single character voice, multiple press key or single press key
- Successfully using RTPengine to enable recordings on the SBC side instead of the PBX
- Write lua code to detect events from FreeSWITCH server and setup that lua code in mod_lua to detect specific header in SIP message and process it with custom dialplan, for example, when receiving SIP message 480 with Reason header cause is 600 and text is voicemail this is a voicemail calls

- Research, apply, document, and train DevOps tools for VoIP Engineer, System Admin, 2 people have become proficient in coding Jenkins pipeline and Ansible playbook to handle my tasks
- Code Jenkins for CI and Ansible for deployment, migrate data, backup, restore, load test, and stress test, some of them can be found on my GitHub repositories and some of them even have a video explain how it works
  - Deploy service API Golang as back-end and NodeJS as front-end developed by the developer team
  - Reduce deployment time VoIP System from 4 hours to 1 hour and reduce human error ▶ Video
  - Centralize the iptables file to manage on the Git server instead of remembering which iptables is the latest on which server ▶ Video
  - Reduce repetitive tasks such as installing zabbix-agent on a VoIP system, logging in to the Zabbix server to add a host, and adding action ▶ Video
  - Migrate one or multiple tenants from a FusionPBX server to another, by generating a bash shell script to dump data of that tenant of 1 server, transfer and restore it on another server
  - Stress test and load test a FusionPBX server using SIPP, Jenkins, and Ansible will install sipp on another server, generating scenario based on Jenkins params and Ansible extra-vars and send SIP message on sipp server to test FusionPBX server ▶ Video
  - Generating script for backup or restore based on what Jenkins parameters the user chooses on Jenkins, Ansible will generate a bash shell script for each service and add a crontab to run daily at a specific time, that script will backup that service to Google Drive via rclone ▶ Video
- All VoIP Engineer and System Admin responsibilities

Nov 2021 — Jun 2019

TEL4VN ICT Training Center

VoIP Engineer, System Admin

Responsible:

- Proactively manage service performance. Provide in-depth analysis of usage trends and make recommendations for modifications to ensure services have sufficient capacity and increase/decrease capacity as needed.
- Operate Zabbix-server as a monitoring server
  - Install zabbix-agent on servers to monitor whether an open port is closed, CPU, RAM, and disk space left
  - Write a custom Zabbix template for monitoring PBX services

- Write a bash shell script to detect concurrent calls of the FreeSWITCH, Asterisk, detect online extensions of the FreeSWITCH, OpenSIPs, and add that script to zabbix-agent UserParameters

- Operate Homer as a SIP capture server
  - Install captagent and config SIP port range on captagent and have it send SIP message log to Homer server to debug calls
  - Integrate Homer to Grafana and Prometheus to show the number of SIP messages sent to Homer by multiple PBX servers by id, and add notify on Grafana to send a notification when any server reaches a breaching point in a period of time, for example, 2 messages SIP 403 - Forbidden continuously for 1 minute will send a notification to email and telegram
- Operate Kamailio as a class 4 Softswitch server with carrier_route module for routing SIP numbers for PBX server
  - Write additional Kamailio code to handle different PSTN provider requirements for From To and Contact header
- Operate OpenSIPs as a SBC (Session Border Control) Server with mid_registrar module and RTPengine for re-write SDP for WebRTC SIP
  - Run between client and PBX server to reduce the rate of Re-REGISTER message
  - WebSocket on OpenSIPs for WebRTC client
  - Meeting with partner companies to help them integrate their system with WebRTC via JsSIP or SIP.js which is a library for web or apps that connect to OpenSIPs WebRTC to use PBX services
- Write a bash shell script for keeping a recording directory of a tenant on the FreeSWITCH server always below the given size or date but with the most recent date (tenant A keeping 5GB of the most recent recordings, tenant B keeping 10GB of the most recent recordings, tenant C keeping last 3 months of the most recent recordings, ...)
- Optimize and tuning database PostgreSQL and MariaDB or MySQL server to maximize concurrent calls and extensions can be run on a PBX server and Autocall server
- Write SIPP code, using SIPP to load test, and stress test with many scenarios to determine how many concurrent calls and extensions can be run on different specs, for example, 2 cores - 2 GB RAM - can run 800 extensions and 200 concurrent calls, 4 cores - 4 GB RAM can run 800 extensions and 200 concurrent calls, ...
- Planning backup and restore solutions when disaster happens
  - Write bash shell script for backup and restore services, dump or restore database PostgreSQL and MariaDB to or from a compressed file, compress or extract all directories for configuration file of asterisk, freeswitch, apache2, nginx, ... to or from a compressed file and upload all of them to a directory on Google Drive via rclone with a organize structure folder by name and date
  - Test restore each quarter to make sure backup files are ready when disaster hit

    - Documented step-by-step command on how to write backup and restore file
- Build and deploy VoIP Systems such as FusionPBX, Autocall Server based on Asterisk, and OpenSIPs on the cloud and on-premise server using a list of step-by-step commands
- Understanding how VoIP Systems running behind NAT should handle NAT
    - Debug calls are disconnected after 30 seconds because of the setup NAT wrong or timeout RTP which is the default 30 second
- Understanding SIP and RTP
    - How a SIP client can "register" to a PBX server
    - Difference between the first REGISTER message and the second REGISTER message to authenticate a SIP client?
    - How expire time of a client works
    - How a SIP client can call each other, negotiate codec and transmit audio to each other
    - How to use sngrep, tcpdump to capture SIP message on PBX server side, run Wireshark on the client to capture SIP message on the client side
    - Differences between SIP dialog and SIP transaction
    - Remember by heart a basic SIP calls flow when client A calls client B via the PBX server. What is their SIP header i.e From To Contact header and what IP:port they are using to send the SIP message, what in SDP i.e what IP:port they are using to send RTP package
- Works with PSTN provider
    - How a CID or DID must be running to send calls to the mobile phone number
    - How outbound calls should send to the PSTN provider server i.e From & To & Contact headers of the SIP message must be
    - How inbound calls are sent to the PBX server i.e PBX server must not send a request for authentication on inbound calls
    - Debug cases of missing audio on 1 side or 2 sides, latency in audio, PSTN provider sending SIP request or response message to the wrong PBX server
- Operate Asterisk as an Autocall server
    - Setup trunk peer with PSTN provider or trunk account with another PBX
    - Write custom dialplan for fail-over calls for each trunk
    - Write custom dialplan by number regex to maximize cost on each mobile phone number
    - Write custom dialplan when calls are connected to the mobile phone number that calls with sent to another PBX server as inbound calls, to minimize the time the user on the PBX server has to call manually
    - Write custom dialplan to transfer autocall calls to another PBX on user response keypress when hearing auto call audio files

    - Meeting and training customers on how to use the Autocall server
- Operate FreeSWITCH as a PBX server
    - Setup multi-tenant with super-admin as the highest permission and admin for managing tenants and creating extensions on each tenant
    - Setup gateway or trunk peer with PSTN provider or trunk account with another PBX
    - Setup outbound calls by regex to maximize cost on each mobile phone number
    - Setup inbound call flow (ring group, call center, IVR, time condition) and consulting customers on how an inbound call flow should work
    - Setup calls cross-tenant for multi-department company
    - Meeting and training customer how to check calls history and calls recordings
- Operate VoIP peripheral
    - Config IP Phone or Softphone for end-users
    - Config and setup GSM Gateway (Simbox or Sim Gateway) to work with Asterisk and FreeSWITCH

Skills

VoIP & Protocols

SIP/RTP/SRTP/WebRTC · STUN/TURN · Asterisk - FreePBX · FreeSWITCH - FusionPBX · GOautodial · Homer · Kamailio · OpenSIPs · RTPengine

System & Observability

Consul · Grafana · Graylog · Loki · OpsGenie · Prometheus/PromQL · Vault · Apache2 · Corosync/Pacemaker · HAProxy · Linux · MariaDB · Nginx · PostgreSQL · Zabbix

DevOps & CI/CD

GitHub Actions · Jinja2 · Ansible · Docker · Git/GitHub · Jenkins · Kubernetes · Terraform · Vagrant

Programming

C/C++ · Bash · Lua · Python

Debugging & Testing

Audio Quality Tools | GDB/Core Dumps | PCAP Analysis | SRTP Decryption | sngrep/tcpdump | SIPp Load Testing | Wireshark/tshark

AI & Automation

Context Engineering | OpenClaw | Telnyx Internal AI

Education

Engineer of Telecommunications

Sai Gon University, Viet Nam

2016 — 2020

Courses & Certificates

Google Cybersecurity
Feb 2026
GOOGLE

IBM Data Engineering
Jan 2026
IBM

Computer Systems and Architecture
Nov 2025
COURSERA

All of AI: ChatGPT, Midjourney, Stable Diffusion & App Dev
May 2025
UDEMY

ASL | First 500+ Basic Signs | American Sign Language
Dec 2024
UDEMY

The C++20 Masterclass: From Fundamentals to Advanced
Apr 2024
UDEMY

AWS Certified Solutions Architect – Associate
Jan 2023
AWS

Google IT Automation with Python
Sep 2022
GOOGLE

Linux Engineer (LPIC-2)
Aug 2022
LPI

Terraform Foundation
Jun 2022
TEL4VN

Kubernetes Administrator
Sep 2021
TEL4VN

Asterisk Certified Essentials
Aug 2021
SANGOMA

Certified Linux Administrator (LPIC-1)
Mar 2021
LPI

Ansible Automation
Jan 2021
TEL4VN

Jenkins Fundamentals
Sep 2020
TEL4VN

LPIC-2: Linux Engineer
May 2020
TEL4VN

Docker Certified Associate
Aug 2019
TEL4VN

LPIC-1: System Administrator
Jun 2019
TEL4VN

895 TOEIC Listening and Reading
Nov 2018
ETS

VoIP Open-source Carrier Certificate
Jul 2018
TEL4VN

VoIP Open-source Administrator Certificate
May 2018
TEL4VN

Languages & Hobbies

Languages

Vietnamese	Native
English	Fluent

Hobbies

- Movies & series
- Music
- Languages

Get in Touch

☎ +84 965 138 057

✉ minhtuan1407.work@gmail.com

in linkedin.com/in/minhtuan1407

</> github.com/minhtuan1407

Ho Chi Minh City, Vietnam