

# Repositorios GitHub uteis para scripts Linux

Pesquisa curada de repositorios com scripts, automacoes, ferramentas de shell, hardening, VPN, homelab, dotfiles e qualidade de Bash. Foco: uso pratico em Linux, nao colecao aleatoria de estrelinhas.

## Resumo honesto

Foram selecionados 52 repositorios. Nem todos sao 'colecoes puras de scripts': alguns sao ferramentas que tornam scripts Linux melhores, mais seguros ou mais produtivos. Isso e intencional. Uma lista so de snippets aleatorios seria menos util e mais perigosa.

Regra de ouro: nunca rode comandos do tipo `curl | bash` em maquina importante sem ler o script, testar em VM/container e ter backup/snapshot. Conveniencia sem auditoria e como dar sudo para um desconhecido simpatico.

## Melhores usos por perfil

Perfil	Repositorios indicados
Seguranca web	acmesh-official/acme.sh
Dev Node/Python	nvm-sh/nvm, pyenv/pyenv
Terminal diario	fzf, gum, Bash-it/Oh My Zsh
Qualidade de scripts	ShellCheck, shfmt, Bats, ShellSpec
VPN	Nyr/openvpn-install, angristan/wireguard-install, hwDSL2/setup-ipsec-vpn
Homelab	community-scripts/ProxmoxVE, DockSTARTer, SimpleHomelab/Docker-Traefik

## Critérios de selecao

Priorizei repositorios que caem em pelo menos uma destas categorias: automacao pratica em Linux, utilidade clara para sysadmin/devops, produtividade real no shell, hardening/auditoria, VPN/rede/DNS, homelab ou qualidade/testes de scripts. Exclui repositorios claramente focados em malware, phishing ou automacao ofensiva sem necessidade administrativa legitima.

# Lista curada

## Certificados e web

### 1. acmesh-official/acme.sh

<https://github.com/acmesh-official/acme.sh>

O que e: Cliente ACME escrito em shell para emitir, renovar e instalar certificados TLS/SSL. Muito bom para servidores Nginx/Apache e automacao de certificados.

Bom para: Automatizar HTTPS em servidores Linux.

Cuidado: Revise os hooks de DNS e reload de servicos antes de automatizar em producao.

## Desenvolvimento

### 2. nvm-sh/nvm

<https://github.com/nvm-sh/nvm>

O que e: Gerenciador de versoes de Node.js implementado como script POSIX shell, com suporte a bash, zsh, dash e WSL.

Bom para: Alternar rapidamente entre versoes de Node por projeto.

Cuidado: Evite instalar Node globalmente via pacote do sistema e nvm ao mesmo tempo sem controlar o PATH.

### 3. pyenv/pyenv

<https://github.com/pyenv/pyenv>

O que e: Gerenciamento simples de multiplas versoes de Python usando shims e shell scripts, sem depender do Python para inicializar.

Bom para: Projetos que exigem Python 3.x diferentes no mesmo Linux.

Cuidado: Instale as dependencias de compilacao da distro antes de compilar versoes novas.

### 4. rbenv/rbenv

<https://github.com/rbenv/rbenv>

O que e: Gerenciador de versoes de Ruby com filosofia pequena e previsivel, baseado em shims.

Bom para: Ambientes Ruby/Rails com versoes diferentes por projeto.

Cuidado: Combine com ruby-build e registre claramente a versao no projeto.

## Shell e produtividade

### 5. Bash-it/bash-it

<https://github.com/Bash-it/bash-it>

O que e: Framework comunitario para Bash com aliases, completions, plugins, temas e scripts auxiliares.

Bom para: Transformar um bashrc baguncado em um setup modular.

Cuidado: Habilite poucos plugins; framework demais tambem vira lentidao no shell.

### 6. ohmyzsh/ohmyzsh

<https://github.com/ohmyzsh/ohmyzsh>

O que e: Framework Zsh com centenas de plugins e temas, mais instalador e atualizador automatico.

Bom para: Terminal Linux mais produtivo, especialmente para git, docker, node, python e kubernetes.

Cuidado: Nao ative plugins sem necessidade; shell lento e erro classico de setup inchado.

## 7. junegunn/fzf

<https://github.com/junegunn/fzf>

O que e: Fuzzy finder de linha de comando com integracoes para bash, zsh, fish, tmux, Vim e Neovim.

Bom para: Buscar arquivos, historico, branches, processos e hosts de forma interativa.

Cuidado: Aprenda a combinar com pipes; o ganho real vem da integracao com seus comandos.

## 8. charmbracelet/gum

<https://github.com/charmbracelet/gum>

O que e: Ferramenta para criar scripts shell interativos e bonitos: menus, confirmacoes, inputs, spinners, tabelas e estilos.

Bom para: Melhorar scripts internos sem escrever TUI do zero.

Cuidado: Otimo para UX, mas ainda precisa de validacao robusta por baixo.

## 9. bpkg/bpkg

<https://github.com/bpkg/bpkg>

O que e: Gerenciador leve de pacotes Bash para instalar scripts globalmente ou por projeto.

Bom para: Distribuir pequenos scripts shell com dependencias simples.

Cuidado: Leia o script instalado; pacote de shell tem poder total no sistema.

# Automacao e scripting

## 10. google/zx

<https://github.com/google/zx>

O que e: Ferramenta do Google para escrever scripts melhores usando JavaScript/TypeScript com chamadas shell mais ergonomicas.

Bom para: Automacoes de build, deploy e tarefas repetitivas em times que ja usam JS.

Cuidado: Nao e bash puro; exige runtime Node e disciplina de dependencias.

# Automacao e parsing

## 11. kellyjonbrazil/jc

<https://github.com/kellyjonbrazil/jc>

O que e: Converte saida de comandos comuns e arquivos para JSON/YAML/dicionarios, facilitando pipelines com jq e scripts.

Bom para: Scripts que precisam processar output de lsblk, ps, dig, df, systemctl etc.

Cuidado: Sempre trate falhas de parsing e diferencas de versao das ferramentas do sistema.

## 12. kellyjonbrazil/jello

<https://github.com/kellyjonbrazil/jello>

O que e: Filtro de JSON e JSON Lines com sintaxe Python, parecido com jq, bom para usuarios que pensam em Python.

Bom para: Pipelines shell que manipulam JSON sem escrever scripts maiores.

Cuidado: Para servidores minimalistas, jq ainda pode ser mais comum por padrao.

# Qualidade de shell

### 13. modernish/modernish

<https://github.com/modernish/modernish>

O que e: Biblioteca para escrever programas POSIX shell mais robustos, portaveis e legiveis.

Bom para: Scripts que precisam rodar em sh/dash/ksh/bash com menos surpresas.

Cuidado: Adiciona dependencia conceitual; vale para scripts maiores, nao para one-liners.

### 14. koalaman/shellcheck

<https://github.com/koalaman/shellcheck>

O que e: Analisador estatico para shell scripts que pega bugs, quoting quebrado, uso inseguro de variaveis e erros comuns.

Bom para: Rodar em todo script .sh antes de colocar em cron, CI ou servidor.

Cuidado: Nao ignore alertas sem entender. Geralmente ele esta certo.

### 15. mvdan/sh

<https://github.com/mvdan/sh>

O que e: Parser, formatter e interpretador de shell; inclui shfmt para formatar scripts de forma consistente.

Bom para: Padronizar estilo de scripts em repositorios e CI.

Cuidado: Formata, mas nao prova que a logica esta correta.

## Testes de scripts

### 16. bach-sh/bach

<https://github.com/bach-sh/bach>

O que e: Framework de testes unitarios para shell scripts.

Bom para: Validar scripts de automacao antes de rodar em servidores reais.

Cuidado: Teste tambem integracao com Docker/VM quando o script altera o sistema.

### 17. shellspec/shellspec

<https://github.com/shellspec/shellspec>

O que e: Framework BDD para shell scripts, com estilo de especificacao e foco em comportamento.

Bom para: Times que querem testes legiveis para scripts complexos.

Cuidado: Nao substitui ShellCheck, shfmt e testes manuais em ambientes reais.

### 18. bats-core/bats-core

<https://github.com/bats-core/bats-core>

O que e: Framework de teste baseado em TAP para Bash, muito usado para validar CLIs e scripts.

Bom para: Criar suites de teste para funcoes bash e comandos.

Cuidado: Isolar estado e filesystem temporario e metade do trabalho.

## Referencias e snippets

### 19. onceupon/Bash-Oneliner

<https://github.com/onceupon/Bash-Oneliner>

O que e: Colecao de one-liners Bash e truques de terminal para processamento de dados e manutencao de sistema Linux.

Bom para: Consulta rapida para grep, awk, sed, xargs, arquivos, processos e manutencao.

Cuidado: One-liner e faca afiada: entenda antes de colar no shell.

## Colecoes de scripts

### 20. davidhcefx/My-Bash-Scripts

<https://github.com/davidhcefx/My-Bash-Scripts>

O que e: Scripts e ferramentas Bash criados para facilitar tarefas do dia a dia no Linux.

Bom para: Inspiracao para utilitarios pessoais e pequenos wrappers.

Cuidado: Trate como biblioteca de ideias; adapte e teste antes de usar.

### 21. miguelgferro/scripts

<https://github.com/miguelgferro/scripts>

O que e: Colecao de scripts shell uteis para Linux, Windows e macOS.

Bom para: Automacoes variadas e referencia de padroes simples.

Cuidado: Por ser multi-plataforma, confira se o script realmente mira sua distro.

### 22. rodmoioliveira/bash-utils

<https://github.com/rodmoioliveira/bash-utils>

O que e: Colecao de shell scripts para Linux Ubuntu e macOS.

Bom para: Utilitarios pessoais e exemplos de automacao Bash.

Cuidado: Leia compatibilidade por distro antes de usar.

### 23. qiray/useful-scripts

<https://github.com/qiray/useful-scripts>

O que e: Colecao de scripts uteis envolvendo git, bash, SSH e automacoes pequenas.

Bom para: Snippets e utilitarios para rotina de terminal.

Cuidado: Repositorio pequeno; trate como ponto de partida.

### 24. pabroux/shell-utilities

<https://github.com/pabroux/shell-utilities>

O que e: Colecao de utilitarios shell de uso geral.

Bom para: Explorar funcoes e pequenos comandos reaproveitaveis.

Cuidado: Valide dependencias externas antes de copiar para seu PATH.

### 25. Olswitcher/bash-scripts

<https://github.com/Olswitcher/bash-scripts>

O que e: Colecao pessoal de scripts Bash usados regularmente.

Bom para: Referencias praticas para scripts do cotidiano.

Cuidado: Script pessoal costuma assumir ambiente do autor; ajuste antes.

### 26. j1i-ian/scripts4u

<https://github.com/j1i-ian/scripts4u>

O que e: Colecao de scripts Bash uteis para evitar ficar garimpando comandos avulsos pela internet.

Bom para: Descobrir snippets e automatizacoes pequenas.

Cuidado: Nao use em producao sem auditoria.

## Sysadmin

### 27. mitchellkrogza/linux-server-administration-scripts

<https://github.com/mitchellkrogza/linux-server-administration-scripts>

O que e: Scripts Bash simples de administracao Linux para facilitar tarefas de servidor.

Bom para: Base para backups, verificacoes e rotinas de administracao.

Cuidado: Projeto antigo; use como referencia, nao como verdade absoluta para sistemas modernos.

### 28. errnair/sysadmin-shell-scripts

<https://github.com/errnair/sysadmin-shell-scripts>

O que e: Scripts Bash e Python para tarefas comuns de administracao Linux em RHEL/Rocky/Alma, Ubuntu e Debian.

Bom para: Automatizar rotina de sysadmin com padroes melhores: set -euo pipefail, logs e backup de configs.

Cuidado: Mesmo com bons padroes, scripts que alteram configs devem rodar primeiro em staging.

### 29. unkn0wnc0de7/sysadmin.sh

<https://github.com/unkn0wnc0de7/sysadmin.sh>

O que e: Script de sysadmin Linux com ferramentas para rede, informacoes de sistema e tarefas de seguranc

Bom para: Menu rapido para diagnostico e administracao em Debian, Arch e Fedora.

Cuidado: Scripts de root com menu sao confortaveis, mas perigosos se voce nao ler o que cada opcao executa.

### 30. Mahmoud7Osman/LSATools

<https://github.com/Mahmoud7Osman/LSATools>

O que e: Conjunto de scripts para administracao de servidores e distribuicoes Linux com menu rapido.

Bom para: Tarefas de administracao recorrentes em servidores Linux.

Cuidado: Menu automation pode esconder comandos destrutivos; leia o codigo.

### 31. t1819/Linux\_Admin\_Scripts

[https://github.com/t1819/Linux\\_Admin\\_Scripts](https://github.com/t1819/Linux_Admin_Scripts)

O que e: Scripts para automatizar trabalhos recorrentes de sysadmin Linux.

Bom para: Rotinas administrativas que se repetem em servidores.

Cuidado: Avalie a qualidade antes de padronizar em ambiente corporativo.

## Sysadmin e web

### 32. Franky5831/IT-Scripts

<https://github.com/Franky5831/IT-Scripts>

O que e: Serie de scripts shell relacionados a TI, Apache/Apache2 e tarefas de administracao.

Bom para: Automacoes de servidor web e operacoes simples.

Cuidado: Revise paths e versoes de pacotes da sua distro.

## Sysadmin e Ubuntu

### 33. pegasusict/PLAT

<https://github.com/pegasusict/PLAT>

O que e: Pegasus Linux Administration Tools: scripts Bash de pos-instalacao e manutencao para Ubuntu.

Bom para: Automatizar setup, manutencao e servicos em ambientes Ubuntu.

Cuidado: Evite rodar em distros nao suportadas sem adaptar.

## Sysadmin e hardening

### 34. EmreOvunc/Linux-System-Management-Scripts-Tricks

<https://github.com/EmreOvunc/Linux-System-Management-Scripts-Tricks>

O que e: Scripts e truques para seguranca, hardening, gerenciamento e configuracao Linux.

Bom para: Referencia de comandos para rede, usuarios, DNS, proxy, RAID, swap e firewall.

Cuidado: Repositorio antigo; use para aprender e revalidar comandos atuais.

## Hardening

### 35. gensecaiHQ/Ubuntu-Security-Hardening-Script

<https://github.com/gensecaiHQ/Ubuntu-Security-Hardening-Script>

O que e: Script de hardening para Ubuntu com foco em OpenSCAP e alinhamento DISA-STIG.

Bom para: Endurecer servidores Ubuntu, especialmente quando ha requisitos de compliance.

Cuidado: Hardening pode quebrar servicos; aplique por etapas com rollback.

### 36. pylllyukko/harden.yml

<https://github.com/pylllyukko/harden.yml>

O que e: Playbook Ansible para hardening Linux em Debian, Slackware e outros ambientes.

Bom para: Aplicar hardening repetivel com Ansible em multiplas maquinas.

Cuidado: Leia cada role/tarefa; hardening generico raramente cabe 100% no seu ambiente.

## Compliance CIS

### 37. ansible-lockdown/UBUNTU24-CIS

<https://github.com/ansible-lockdown/UBUNTU24-CIS>

O que e: Automacao Ansible para remediacao de baseline CIS em Ubuntu 24.

Bom para: Ambientes que precisam aproximar Ubuntu 24 de controles CIS.

Cuidado: Rode primeiro em auditoria/staging; CIS pode entrar em conflito com aplicacoes.

### 38. ansible-lockdown/DEBIAN12-CIS

<https://github.com/ansible-lockdown/DEBIAN12-CIS>

O que e: Automacao Ansible para remediacao CIS em Debian 12.

Bom para: Padronizar hardening de Debian 12 em frota.

Cuidado: Nao use como botao magico de seguranca; documente excecoes.

## Auditoria e seguranca

### 39. CISOfy/lynis

<https://github.com/CISOfy/lynis>

O que e: Ferramenta de auditoria de segurança para Linux, Unix e macOS, muito usada para checagens de hardening.

Bom para: Auditar hosts, descobrir fraquezas e gerar recomendações antes/depois de hardening.

Cuidado: Recomendacao nao e obrigacao; priorize pelo risco real do ambiente.

## VPN e rede

### 40. Nyr/openvpn-install

<https://github.com/Nyr/openvpn-install>

O que e: Instalador OpenVPN road warrior para Ubuntu, Debian, AlmaLinux, Rocky Linux, CentOS e Fedora.

Bom para: Subir OpenVPN rapido em VPS ou servidor dedicado.

Cuidado: Nao rode em desktop pessoal; use servidor limpo e revise firewall.

### 41. Nyr/wireguard-install

<https://github.com/Nyr/wireguard-install>

O que e: Instalador WireGuard road warrior para varias distros Linux.

Bom para: Criar VPN WireGuard simples e gerenciar usuarios depois.

Cuidado: Proteja chaves privadas e revise regras NAT/iptables.

### 42. angristan/wireguard-install

<https://github.com/angristan/wireguard-install>

O que e: Script Bash para instalar, configurar e gerenciar WireGuard em servidores Linux.

Bom para: VPN com IPv4/IPv6, systemd e configuracoes de cliente.

Cuidado: Rode em servidor dedicado/VPS, nao em maquina de trabalho.

### 43. hwdsl2/setup-ipsec-vpn

<https://github.com/hwdsl2/setup-ipsec-vpn>

O que e: Scripts para configurar servidor IPsec/L2TP, Cisco IPsec e IKEv2 em minutos, com scripts auxiliares de usuarios e certificados.

Bom para: VPN IPsec compativel com varios clientes nativos.

Cuidado: Leia os avisos do projeto: e para servidores, nao para PC/Mac pessoal.

## VPN e containers

### 44. hwdsl2/docker-ipsec-vpn-server

<https://github.com/hwdsl2/docker-ipsec-vpn-server>

O que e: Imagem e scripts para rodar servidor IPsec VPN em Docker.

Bom para: Ambientes que preferem containerizar VPN IPsec.

Cuidado: Networking de Docker + VPN e delicado; teste rotas, NAT e firewall.

## DNS e privacidade

## 45. pi-hole/pi-hole

<https://github.com/pi-hole/pi-hole>

O que e: Projeto Pi-hole com instalador automatizado e scripts shell para DNS sinkhole e bloqueio de anuncios na rede.

Bom para: Bloqueio de ads/tracking em rede domestica ou laboratorio.

Cuidado: DNS central quebrado derruba navegacao da rede; tenha plano de rollback.

## Homelab e Proxmox

### 46. community-scripts/ProxmoxVE

<https://github.com/community-scripts/ProxmoxVE>

O que e: Scripts comunitarios para instalacao em um comando de servicos, containers e VMs no Proxmox VE.

Bom para: Acelerar homelab com LXC/VMs para automacao, midia, redes, bancos, monitoramento etc.

Cuidado: Mesmo sendo popular, script de Proxmox mexe em infraestrutura. Snapshot antes, sempre.

## Docker e homelab

### 47. GhostWriters/DockSTARTer

<https://github.com/GhostWriters/DockSTARTer>

O que e: Ferramenta para subir rapidamente stacks Docker, gerar configuracoes Compose e simplificar setup de containers.

Bom para: Servidor Linux domestico ou laboratorio com varios containers.

Cuidado: Nao edite arquivos gerados no impulso; entenda overrides e backups.

### 48. SimpleHomelab/Docker-Traefik

<https://github.com/SimpleHomelab/Docker-Traefik>

O que e: Repositorio real de Docker Compose com Traefik, CrowdSec, Authelia/OAuth2, LetsEncrypt e muitos apps de homelab.

Bom para: Aprender arquiteturas reais de reverse proxy, auth e servicos self-hosted.

Cuidado: Nao copie stack inteira sem mapear dominios, secrets, volumes e firewall.

## Docker e containers

### 49. linuxserver/docker-mods

<https://github.com/linuxserver/docker-mods>

O que e: Documentacao e exemplos para criar mods de containers LinuxServer.io adicionando arquivos e scripts no boot.

Bom para: Customizar containers LinuxServer sem manter forks completos.

Cuidado: Mods rodam dentro do container; valide ordem de init e impacto nas atualizacoes.

## Dotfiles e bootstrap

## 50. irichu/dotfiles

<https://github.com/irichu/dotfiles>

O que e: Dotfiles multiplataforma com instalador para Ubuntu Desktop, servidores Ubuntu/Fedora/Arch Linux, macOS e Termux.

Bom para: Estudar bootstrap moderno de terminal, zsh, starship, tmux, Neovim e ferramentas CLI.

Cuidado: Dotfiles de outra pessoa podem sobrescrever habitos e configs. Fork antes.

## 51. yadm-dev/yadm

<https://github.com/yadm-dev/yadm>

O que e: Yet Another Dotfiles Manager: gerenciador de dotfiles baseado em Git, com bootstrap, hooks, templates e criptografia.

Bom para: Sincronizar configuracoes Linux entre maquinas de forma controlada.

Cuidado: Nao versionar segredos sem criptografia e revisao de .gitignore.

## 52. twpayne/chezmoi

<https://github.com/twpayne/chezmoi>

O que e: Gerenciador poderoso de dotfiles com templates, segredos e suporte a multiplas maquinas.

Bom para: Padronizar setup pessoal/profissional em varios Linux sem copiar arquivos manualmente.

Cuidado: Templates poderosos tambem criam complexidade. Mantenha simples.

# Como usar esta lista sem se ferrar

1. Clone o repositório e leia os scripts principais antes de executar. 2. Rode ShellCheck/shfmt nos scripts que você adaptar. 3. Teste em VM, container ou máquina descartável. 4. Em servidores, tire snapshot/backup antes. 5. Controle versão dos seus ajustes. 6. Evite scripts abandonados para mudanças sensíveis de segurança, kernel, rede ou firewall.

Minha recomendação prática: transforme os repos desta lista em uma pasta de laboratório. Nunca trate GitHub como package manager de produção sem pin de versão, auditoria e rollback.

## Fontes consultadas

A pesquisa combinou páginas de tópicos do GitHub, repositórios específicos e documentações relacionadas. Links abaixo foram usados para validar descrições e selecionar categorias.

- GitHub Topics - bash-script: <https://github.com/topics/bash-script>
- GitHub Topics - useful-bash-scripts: <https://github.com/topics/useful-bash-scripts>
- GitHub Topics - shell-scripting: <https://github.com/topics/shell-scripting>
- GitHub Topics - linux-administration: <https://github.com/topics/linux-administration>
- GitHub Topics - linux-hardening: <https://github.com/topics/linux-hardening>
- Dotfiles utilities: <https://dotfiles.github.io/utilities/>
- acme.sh README: <https://github.com/acmesh-official/acme.sh>
- nvm README: <https://github.com/nvm-sh/nvm>
- pyenv README: <https://github.com/pyenv/pyenv>
- Bash-it README: <https://github.com/Bash-it/bash-it>
- Oh My Zsh README: <https://github.com/ohmyzsh/ohmyzsh>
- gum README: <https://github.com/charmbracelet/gum>
- setup-ipsec-vpn README: <https://github.com/hwds12/setup-ipsec-vpn>
- ProxmoxVE Helper-Scripts: <https://github.com/community-scripts/ProxmoxVE>
- Pi-hole install docs: <https://docs.pi-hole.net/main/basic-install/>
- DockSTARTer docs: <https://dockstarter.com/>

Gerado em 26/04/2026. Recomendação final: antes de executar scripts de terceiros, leia, teste e registre rollback.