



*EMENTA*

# LINUX PARA CLOUD NATIVE



TREINAMENTO EM TURMA

# LINUX PARA CLOUD NATIVE



DESENVOLVIMENTO INFRAESTRUTURA CLOUD DEVOPS

6 MESES INICIANTE

este é um treinamento totalmente prático e moderno, desenhado para eliminar a lacuna entre o desenvolvimento e a infraestrutura.

o treinamento cobre desde a perda do medo do terminal (“tela preta”) até a automação de deploys na nuvem. utilizaremos uma metodologia híbrida: o aluno aprenderá simultaneamente em um ambiente local (virtualbox) e em um ambiente de nuvem real (aws ec2), cobrindo ssh, permissões, manipulação de arquivos, systemd, gerenciamento de processos, storage/discos, networking e introdução a containers com docker.

ACESSO POR  
36 MESES



MATERIAL  
DE APOIO



AULAS  
PRÁTICAS



SUPORTE



## PRA QUEM É ESSE TREINAMENTO

**DESENVOLVEDORES (DEVS)** que querem parar de depender de infra para subir suas aplicações e desejam migrar para devops.

**SYSADMINS TRADICIONAIS** que precisam se atualizar urgentemente para o mundo de cloud e containers.

**ASPIRANTES A DEVOPS/SRE** quem está começando do zero e busca a base sólida para aprender kubernetes depois.

**PROFISSÕES ALVO** DEVOPS ENGINEER, CLOUD ANALYST, SITE RELIABILITY ENGINEER (SRE), PLATFORM ENGINEER, DEV.

### PRÉ-REQUISITOS

- ☑ computador (WINDOWS, MAC OU LINUX) com pelo menos 8gb de ram (para rodar uma vm leve).
- ☑ não é necessário conhecimento prévio em LINUX.
- ☑ conhecimentos básicos em informática.

# GRADE DE APRENDIZADO





## AULA 1

### └ o setup híbrido & ssh

este módulo cobre a mentalidade cloud native (por que linux domina a nuvem), a diferença entre virtualização local e instâncias cloud, a montagem de dois laboratórios práticos (*virtualbox + aws ec2*) e o domínio do ssh como ferramenta principal de trabalho. você aprende sobre key pairs, proteção de chaves *.pem*, configuração do *~/.ssh/config* para simplificar conexões e diagnóstico de problemas de acesso. sai com dois ambientes funcionando e conectividade segura entre eles.

## AULA 2

### └ sobrevivência no shell

aprenda o sistema de arquivos LINUX (FHS) e a filosofia “tudo é arquivo”. cobre navegação com *PWD*, *CD* e *LS -LAH*, decomposição completa de cada coluna da saída do *LS -L* (tipo, permissões, links, dono, grupo, tamanho, data), manipulação de arquivos e diretórios (*mkdir -p*, *cp*, *mv*, *rm*), wildcards e globs, e os atalhos de produtividade que separam o amador do profissional: *TAB*, *CTRL+R*, *HISTORY*, aliases e atalhos de terminal.



### AULA 3

## └ edição de texto (do VIM ao VS CODE)

este módulo cobre visualização de arquivos (CAT, LESS, HEAD, TAIL -F), o vim em modo sobrevivência (modos normal, insert e visual, fluxo básico de edição, busca com /, substituição com :%s, configuração do .VIMRC), e o VS CODE REMOTE SSH para edição com conforto (instalação da extensão, conexão à EC2, terminal integrado). inclui DIFF para comparar arquivos e a disciplina do backup antes de editar.

### AULA 4

## └ permissões e segurança (users & groups)

aprenda a hierarquia de poder (ROOT vs usuários comuns), criação e gestão de usuários (*adduser, useradd*) e grupos (*groupadd, usermod -ag*), o mecanismo do sudo e a edição segura do */etc/sudoers* com visudo. detalha permissões de arquivos (rwx) em notação octal e simbólica, decomposição completa da saída *drwxr-xr-x*, propriedade com *chown* e *chgrp*, o *umask* e como ele define permissões padrão, o sticky bit em diretórios compartilhados, e hardening do ssh (desabilitar root login, senha, e configurar acesso por chave).



## AULA 5

### └ gerenciamento de pacotes e systemd

este módulo cobre o gerenciador de pacotes apt (repositórios, */etc/apt/sources.list*, instalação, atualização, remoção, *apt search*, *dpkg*), o systemd como pid 1 e gerente de serviços (o que é uma unit, anatomia de um *.service*), o ciclo de vida dos serviços (*systemctl start/stop/restart/reload/enable/disable*), a diferença entre *restart* e *reload*, *systemctl is-active/is-enabled* para scripts, e análise de logs com *journalctl* (filtros por tempo, unidade, prioridade e boot).

## AULA 6

### └ processos e monitoramento

este módulo cobre como o kernel gerencia processos (fork, PID, PPID, a árvore de processos desde o PID 1), estados de processos (RUNNING, SLEEPING, ZOMBIE, STOPPED), monitoramento com *ps aux* (decomposição de cada coluna), *top*, *htop* e *uptime* (LOAD AVERAGE explicado de verdade), sinais unix (SIGTERM, SIGKILL, SIGHUP, SIGSTOP, SIGCONT), o */proc* como interface do kernel, processos em background (*&*, *jobs*, *fg*, *bg*), o OOM KILLER e o que acontece quando a ram acaba, *nohup* vs *tmux* para manter processos vivos após desconectar do ssh.



## AULA 8

### └ streams, pipes e filtros

este módulo cobre os standard streams (stdin/stdout/stderr e seus file descriptors), redirecionamento ( $>$ ,  $>>$ ,  $2>$ ,  $2>&1$ ,  $&>$ ), o pipe como mecanismo de composição, e os filtros essenciais: *grep* (com *-i*, *-v*, *-r*, *-c*, *-n*), *wc*, *sort*, *uniq* (sempre com *sort* antes), *cut*, *awk* (extração de colunas e formatação), *tr* (substituição de caracteres), *tee* (escrever em arquivo e stdout ao mesmo tempo, com *sudo*), e *xargs* (transformar linhas em argumentos). inclui pipelines de análise de logs do mundo real.

## AULA 9

### └ processos e monitoramento

veja endereçamento IP com *ip addr* (decomposição completa), *ip* privado vs público (com diagrama de vpc aws), *dns* local (*/etc/hosts*, */etc/resolv.conf*, *dig*), protocolos TCP/UDP e portas conhecidas, *ss -tulpn* para inspeção de conexões (diferença entre 0.0.0.0 e 127.0.0.1), teste de conectividade com *ping* e *nc*, o *curl* como canivete suíço (headers, status codes, *-v*, apis rest, medição de tempo), firewalls em duas camadas (*SECURITY GROUPS* na AWS e *ufw* no LINUX, com diagrama do modelo), e ferramentas de diagnóstico de rota (*traceroute*, *mtr*). inclui um fluxograma completo de troubleshooting de conectividade.

## AULA 10

### └ automação com shell script

veja fundamentos do bash (shebang, chmod +x, command substitution com `$()`), variáveis (definição, `$1/$@/$#` para argumentos, `read` para input interativo, variáveis de ambiente e `$path`), estruturas de controle (if/elif/else com operadores numéricos, strings e arquivos, for e while), tratamento de erros profissional ( `$?`, `&&/||`, `set -e`, `pipefail`, `trap exit`), funções com local e return. inclui scripts reais de automação: backup com validação e limpeza, health check

com auto-restart de serviços, provisionamento de servidor e limpeza inteligente de logs. cobre agendamento com cron (formato, armadilhas de path) e systemd timers (persistent=true), e boas práticas como idempotência e shellcheck.



## AULA 11

### └ LINUX para containers & projeto final

o que são containers (namespaces PID/NET/MNT, CGROUPS, OVERLAYFS), a diferença fundamental entre container e VM (diagrama), instalação correta do DOCKER pelo repositório oficial (nunca apt install docker.io), permissionamento com usermod -ag docker, DOCKER run com todas as flags essenciais (-d, -p, --name, -v, -e), bind mounts (diretório do host no container, :ro), DOCKER volumes (dados persistentes gerenciados), networking de containers (bridge, redes nomeadas, DNS interno), docker compose para múltiplos serviços,

DOCKERFILES para imagens personalizadas (com ALPINE para imagens menores), e monitoramento de containers com ferramentas LINUX nativas (docker stats, /proc/pid).

#### **o projeto final integra tudo:**

provisionar EC2, aplicar hardening de segurança (usuário deployer, ssh, ufw), instalar docker, criar script de deploy automatizado (deploy.sh) que baixa imagem, sobe container com volume persistente e verifica saúde, e validar acessando o IP público para ver a aplicação rodando.



QUEM SERÁ  
SEU GUIA:



INSTRUTOR

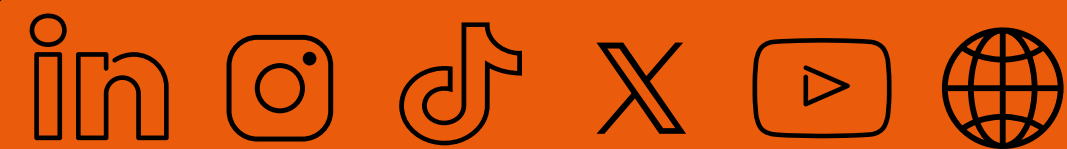
# JEFERSON FERNANDO

**PRINCIPAL ENGINEER**  
NO IFOOD



## SOBRE O JEFERSON

apaixonado por software livre e pela democratização do conhecimento, JEFERSON soma mais de 20 anos de experiência em ambientes de alta criticidade, com passagens por gigantes como WALMART e banco SANTANDER. há uma década, fundou um canal no YOUTUBE que hoje conta com mais de 110 mil inscritos e 700 vídeos, que se tornou a LINUX TIPS. o projeto nasceu com o propósito social de levar conteúdos sobre LINUX, KUBERNETES e DEVOPS a quem não possui recursos para cursos pagos ou enfrenta barreiras com o idioma inglês. sua atuação vai além das telas, tendo palestrado em mais de 50 eventos pelo BRASIL. é um ferrenho defensor da diversidade na tecnologia, liderando iniciativas de impacto real.



*SOBRE A*

# LINUXTIPS

**DESDE 2014 COMPARTILHANDO  
CONHECIMENTO E SENDO A RE-EVOLUÇÃO  
NA FORMA DE APRENDER SOBRE TI**

**hoje, a LINUXTIPS mantém mais de 20  
treinamentos atualizados:** todos  
focados na prática. acreditamos que  
a melhor forma de aprender algo em  
tecnologia é reproduzindo, testando,  
errando, corrigindo e evoluindo rápido.

*FALE CONOSCO.*

**contato@linuxtips.io**



sabemos do nosso papel na sociedade  
em ajudar a levar conhecimento a todos,  
e sendo assim, não fugimos de nossas  
responsabilidades e **estamos sempre  
atuando de perto nas comunidades.**

