

Perfeito 🎉 Abaixo está o guia já no **seu template .md** (com âncoras, emojis, comandos separados em blocos **bash** e assinatura no final).

🚀 Guia Rápido – Pós-instalação do Rocky Linux 10 Minimal

📄 Resumo

Este guia reúne as principais ferramentas e repositórios que devem ser instalados logo após a configuração ****minimal**** do Rocky Linux 10. Inclui: repositórios extras, ferramentas de rede, compilação, containers, segurança, firewall, virtualização e otimizações de shell.

📖 Sumário

- [🔄 Atualizar sistema e repositórios](#-atualizar-sistema-e-repositórios)
- [🔧 Pacotes base](#-pacotes-base)
- [🌐 Rede e diagnóstico](#-rede-e-diagnóstico)
- [⚙️ Compilação e toolchain](#-compilação-e-toolchain)
- [🔒 SELinux e Firewall](#-selinux-e-firewall)
- [🕒 Tempo e timezone](#-tempo-e-timezone)
- [💻 Virtualização (KVM/libvirt)](#-virtualização-kvmlibvirt)
- [📦 Containers (Podman/Buildah/Skopeo)](#-containers-podmanbuildahskopeo)
- [🌟 Extras úteis](#-extras-úteis)
- [✅ Checklist final](#-checklist-final)

🔄 Atualizar sistema e repositórios

```
```bash
sudo dnf -y update
sudo dnf -y install dnf-plugins-core
sudo dnf config-manager --set-enabled crb
sudo dnf -y install epel-release
```

---

## 🔧 Pacotes base

```
sudo dnf -y install vim-enhanced bash-completion man-db man-pages less tree
which tar zip unzip bzip2 xz pigz p7zip p7zip-plugins wget curl rsync openssh-
clients ca-certificates
```

---

## 🌐 Rede e diagnóstico

```
sudo dnf -y install iproute iputils net-tools ethtool nmap-ncat nmap tcpdump
traceroute mtr bind-utils socat telnet lsof strace htop iotop perf
policycoreutils setools-console
```

---

## ⚙️ Compilação e toolchain

```
sudo dnf -y groupinstall "Development Tools"
```

```
sudo dnf -y install gcc gcc-c++ make cmake ninja-build pkgconf pkgconf-pkg-
config autoconf automake libtool kernel-headers kernel-devel elfutils-libelf-
devel openssl-devel zlib-devel libffi-devel bison flex python3 python3-pip
```

---

## 🔒 SELinux e Firewall

### ▽ Desabilitar SELinux

```
sudo vim /etc/selinux/config
```

Altere:

```
SELINUX=disabled
```

Depois reinicie:

```
sudo reboot
```

## 💧 Firewalld

```
sudo dnf -y install firewalld
sudo systemctl enable --now firewalld
```

## Exemplos de liberações

### Por serviço:

```
sudo firewall-cmd --add-service=http --permanent
sudo firewall-cmd --add-service=https --permanent
```

```
sudo firewall-cmd --add-service=ssh --permanent
sudo firewall-cmd --add-service=cockpit --permanent
sudo firewall-cmd --reload
```

### Por porta:

```
sudo firewall-cmd --add-port=8080/tcp --permanent
sudo firewall-cmd --reload
```

### Por protocolo:

```
sudo firewall-cmd --add-protocol=icmp --permanent
sudo firewall-cmd --reload
```

---

## Tempo e timezone

```
sudo dnf -y install chrony
sudo systemctl enable --now chronyd
sudo timedatectl set-timezone America/Sao_Paulo
```

---

## Virtualização (KVM/libvirt)

```
sudo dnf -y install qemu-kvm libvirt virt-install virt-viewer libvirt-client
sudo systemctl enable --now libvirtd
```

---

## Containers (Podman/Buildah/Skopeo)

```
sudo dnf -y install podman buildah skopeo crun
```

---

## Extras úteis

```
sudo dnf -y install git jq yq parted lvm2 smartmontools atop ncd
```

## ☑ Checklist final

```
dnf check-update
dnf repolist
gcc --version
getenforce # deve retornar Disabled
sudo firewall-cmd --list-all
chronyc sources -v
virsh list --all
podman info
```

Bora deixar o **autocompletion** redondo no Rocky Linux (bash). Vou cobrir instalação, habilitação global e alguns atalhos úteis (kubectl/helm/oc, etc.).

### 1) Instalar o pacote base

Instala o mecanismo de auto-complete do bash:

```
sudo dnf install -y bash-completion
```

### 2) Habilitar para **todos os usuários** (system-wide)

Crie o script em `/etc/profile.d/` que carrega o bash-completion ao abrir um shell:

```
sudo vim /etc/profile.d/bash_completion.sh
```

Dentro do arquivo, coloque exatamente:

```
Habilita bash-completion para shells interativos
if [-n "$PS1"]; then
 if [-r /usr/share/bash-completion/bash_completion]; then
 . /usr/share/bash-completion/bash_completion
 fi
fi
```

Salve no **vim** e garanta permissões padrão:

```
sudo chmod 644 /etc/profile.d/bash_completion.sh
```

### 3) Habilitar no **usuário atual** (fallback)

Se quiser garantir mesmo sem re-login, adicione ao seu `~/ .bashrc`:

```
grep -q 'bash-completion' ~/.bashrc || echo '[-r /usr/share/bash-completion/bash_completion] && . /usr/share/bash-completion/bash_completion' >> ~/.bashrc
```

## 4) Recarregar o shell

Aplique sem deslogar:

```
exec bash
ou
source ~/.bashrc
```

## 5) Validar que está ativo

Estes testes devem funcionar sem erro:

```
type _init_completion
complete -p | head
```

Dica rápida: digite `dnf rep<Tab>` ou `systemctl res<Tab>` e veja o preenchimento.

---

## (Opcional) Compleções de ferramentas populares

Se você usa Kubernetes e companhia, registre as entradas em `/etc/bash_completion.d/`:

### kubectl

```
kubectl completion bash | sudo tee /etc/bash_completion.d/kubectl > /dev/null
```

### helm

```
helm completion bash | sudo tee /etc/bash_completion.d/helm > /dev/null
```

### oc (OpenShift)

```
oc completion bash | sudo tee /etc/bash_completion.d/oc > /dev/null
```

### awscli

```
echo 'complete -C "$(command -v aws_completer)" aws' | sudo tee
/etc/bash_completion.d/aws > /dev/null
```

**terraform** (faz tudo sozinho para o usuário atual)

```
terraform -install-autocomplete
exec bash
```

**docker-compose** (se o pacote já trouxe o arquivo, garanta o link no lugar padrão)

```
if [-f /usr/share/bash-completion/completions/docker-compose]; then
 sudo ln -sf /usr/share/bash-completion/completions/docker-compose
/etc/bash_completion.d/docker-compose
fi
```

---

## Notas rápidas

- Em Rocky 8/9, **só instalar bash-completion** já costuma habilitar para sessões novas. O passo do `/etc/profile.d/` garante de forma explícita e padronizada no servidor.
- Se você usa **zsh**, o mecanismo é outro (compinit). A gente configura depois se quiser.

Quer que eu gere um mini-script para auditar e instalar as compleções que você escolher (kubectl/helm/oc/aws/terraform) tudo de uma vez?

---

 Criado por **Jefferson Salles**  [LinkedIn](#)  [jefersonmattossalles@gmail.com](mailto:jefersonmattossalles@gmail.com)

---

Quer que eu adicione também uma seção **\*\*💡 Pós-reboot\*\*** com dicas de ajuste de ``sysctl``, ``fstrim.timer`` e ``tuned`` (performance) para deixar o servidor pronto para produção?