

## Resumo do livro Curso Linux Essentials e lista e exercícios

**Tópico 1:** O Linux, criado por Linus Torvalds em 1991, é um sistema operacional baseado no modelo de código aberto, referindo-se principalmente ao kernel, o núcleo do sistema. Originado do projeto GNU, iniciado por Richard Stallman em 1983 para desenvolver um sistema de código aberto similar ao Unix, o Linux e suas distribuições oferecem uma alternativa robusta e flexível aos sistemas proprietários. As distribuições Linux, como Debian, Ubuntu e CentOS, incluem o kernel Linux e programas GNU, além de ferramentas para facilitar a instalação e gestão de pacotes. Uma das vantagens do Linux é o seu custo, com muitas distribuições gratuitas e a possibilidade de personalização para diferentes necessidades.

O Linux também se adapta a diversas arquiteturas, como x86 e ARM, sendo utilizado em dispositivos variados, desde servidores até smartphones, como o Android. Além disso, o Linux é essencial em sistemas embarcados, como o Raspberry Pi, e em tecnologias emergentes, como a Internet das Coisas (IoT). A computação em nuvem também utiliza distribuições Linux, oferecendo recursos sob demanda com escalabilidade e segurança.

Quanto a aplicativos, o Linux oferece diversas opções de software livre para funções variadas, como suítes de escritório. O LibreOffice, um dos pacotes mais populares, é compatível com os formatos do Microsoft Office e é amplamente utilizado nas distribuições Linux. Com seu modelo de código aberto, o Linux continua a ser uma escolha poderosa e acessível para uma variedade de usuários e dispositivos.

**Tópico 2:** A linha de comando no Linux é uma interface tradicional de interação, especialmente útil em servidores de rede. O shell, como o Bash, interpreta os comandos digitados pelo usuário e os executa. O prompt do shell é indicado pelo caractere \$ (usuário comum) ou # (usuário root). O usuário root tem permissões totais no sistema, podendo modificar qualquer arquivo ou tarefa, ao contrário dos usuários comuns que possuem restrições.

Comandos no shell podem ter opções e argumentos que alteram seu comportamento e especificam os alvos da ação. Por exemplo, o comando echo exibe textos na tela, com opções como -n (sem quebra de linha) e -e (para caracteres especiais). Comandos longos podem ser quebrados com a barra invertida (\).

As variáveis no shell armazenam valores, como caminhos e configurações do sistema, e podem ser exibidas com o prefixo \$. A variável **PATH**, por exemplo, define diretórios onde os programas são procurados. Comandos podem ser interrompidos com **Ctrl + C** se demorarem muito. Para obter ajuda, os comandos man e info fornecem manuais detalhados, enquanto o comando which encontra a localização de programas. A autocompletação é facilitada pela tecla Tab.

**Tópico 3:** No Linux, a manipulação de arquivos pode envolver compressão, armazenamento e busca de informações. O comando **tar** é usado para agrupar arquivos e diretórios, como o comando tar cvf etc.tar /etc, que cria um arquivo compactado. Para descompactá-lo, usa-se tar xvf etc.tar. A compressão adicional pode ser feita com **gzip** ou **bzip2**, como em gzip etc.tar ou bzip2 etc.tar, que resultam em arquivos .tar.gz ou .tar.bz2, respectivamente. O comando **zip** também pode ser utilizado para criar arquivos compactados, com a opção zip documentos.zip \*.pdf.

Para examinar conteúdos de arquivos de texto, o comando **less** permite visualizar e pesquisar por texto em arquivos grandes. O **grep** facilita a busca em arquivos usando expressões regulares, como grep '^..tp \*.../tcp' para procurar serviços de rede. O redirecionamento de dados também

é comum, onde a saída de um comando pode ser direcionada para um arquivo usando `>`, ou redirecionada para outros comandos com `|` (canalização).

Além disso, comandos como **cat**, **wc**, **cut**, **head**, **tail**, **sort**, **paste** e **tac** são usados para manipular, contar e filtrar conteúdo de arquivos. Essas ferramentas são frequentemente combinadas com redirecionamentos e canalizações para realizar tarefas complexas de forma eficiente.

**Tópico 4:** Os sistemas operacionais mais comuns são Windows, Mac OS X e Linux, cada um com suas vantagens e desvantagens. O Windows é amplamente utilizado devido à sua familiaridade e compatibilidade com diversos dispositivos, mas é restritivo e vulnerável a malwares. O Mac OS X oferece um desempenho integrado e uma interface estética, mas é menos compatível com hardware de outros fabricantes. O Linux é livre, versátil e ideal para usuários avançados, mas pode apresentar falta de padronização entre suas distribuições e suporte limitado. Em termos de hardware, periféricos como USB, HDMI e Ethernet são comuns, e a placa-mãe conecta os componentes internos. O Linux utiliza um sistema de arquivos hierárquico (FHS), com diretórios específicos para diferentes funções, como `/bin` para programas essenciais e `/home` para os diretórios dos usuários. A administração de processos e recursos do sistema é feita por ferramentas como `top` e `free`.

**Tópico 5:** O Linux é um sistema operacional multiusuário, permitindo que várias pessoas utilizem o sistema de maneira independente, com cada uma tendo acesso restrito aos seus próprios arquivos e configurações. O usuário `root` é o administrador do sistema e tem acesso irrestrito a todos os recursos e arquivos, sendo fundamental para a manutenção e configuração do sistema. Embora o `root` tenha poderes amplos, é recomendável que tarefas diárias sejam feitas com uma conta de usuário comum, utilizando o comando `sudo` para executar comandos administrativos quando necessário.

No Linux, usuários são identificados por um UID (User ID), e podem ser associados a um ou mais grupos, cada grupo com um GID (Group ID). O arquivo `/etc/passwd` armazena informações básicas dos usuários, como nome de login, UID, GID, e diretório pessoal. Para maior segurança, as senhas dos usuários são armazenadas no arquivo `/etc/shadow`, em formato criptografado, acessível apenas ao `root`.

A criação de novos usuários é feita com o comando `useradd`, que também permite configurar diretórios e grupos para os usuários. O administrador também pode alterar informações das contas, como senha, descrição e shell padrão, utilizando o comando `usermod`. Para controlar o acesso de um usuário a um grupo específico, pode-se utilizar comandos como `gpasswd` e `groupadd`, que permitem associar ou excluir usuários de grupos.

Além disso, o Linux oferece mecanismos para bloquear e desbloquear contas de usuário com o comando `passwd`, além de permitir a exclusão de contas com `userdel`. O sistema de grupos também facilita a organização de permissões, garantindo que apenas usuários autorizados possam acessar arquivos restritos. Com esses recursos, o Linux oferece um alto nível de segurança e flexibilidade para gerenciar usuários e grupos.

## Atividades Teóricas

### Instruções:

- A resposta deve ser dada logo abaixo da pergunta;
- Responda as perguntas de maneira resumida;
- Responda APENAS o que foi perguntado;
- Faça as observações que achar necessário nos comentários;

#### 1. O que é um sistema operacional?

É um conjunto de programas que gerencia os recursos de hardware e software de um computador ou dispositivo.

#### 2. Qual o objetivo do Projeto GNU?

Tinha objetivo de criar um sistema operacional livre e de código aberto, que fosse completamente baseado em software livre, ou seja, software que qualquer pessoa poderia usar, modificar e distribuir livremente.

#### 3. O que é código aberto?

Refere-se a um tipo de software cujo código fonte está disponível publicamente e pode ser livremente acessado, modificado, distribuído e utilizado por qualquer pessoa, deixando dessa forma, o processo mais fácil, prático e flexível.

#### 4. O que são as Distribuições Linux?

As distribuições Linux são variantes do sistema operacional Linux que atendem a diferentes necessidades e preferências dos usuários. Elas variam em termos de facilidade de uso, objetivos e pacotes de software incluídos, mas todas compartilham o mesmo núcleo (kernel) do Linux, proporcionando uma base comum para diversas experiências computacionais.

#### 5. O que é computação em nuvem?

É uma forma de prover serviços em que o sistema operacional fica armazenado em um servidor em que o cliente não tem a necessidade de gerenciar fisicamente o hardware ou o software.

#### 6. O que é NFS?

é um protocolo de compartilhamento de arquivos em rede que permite que sistemas de computadores compartilhem arquivos e diretórios entre si, de forma transparente, como se estivessem localmente em cada máquina.

7. Quais são os principais pontos defendidos por aqueles que acreditam em código fechado?

Defensores do código fechado acreditam que este modelo oferece maior controle sobre a qualidade, segurança, suporte, inovação e monetização do software.

8. Quais são os principais pontos defendidos por aqueles que acreditam em código aberto?

Defensores do código aberto acreditam que ele oferece maior liberdade, transparência, dinamicidade e colaboração.

9. De que formas são remunerados os desenvolvedores de programas de código aberto (inclusive empresas)?

Existem várias formas, dentre elas o financiamento coletivo, patrocínio de empresas, vendas de serviços, consultorias e outros.

10. Quais são as 4 liberdades principais da GLP?

As 4 liberdades são: liberdade de usar o software para qualquer propósito, liberdade para estudar e modificar-lo, liberdade para redistribuir o software e distribuir versões modificadas.

11. Quais são os outros tipos de licença oferecidas pela Free Software Foundation?

GPL, LGPL, MIT e Apache.

12. Qual o principal objetivo da OSI?

Promove e protege o uso de software aberto, realizando a promoção e certificação de licenças de código aberto, assegurando que o software liberado sob essas licenças esteja em conformidade com os princípios de software livre e aberto.

13. Quais os critérios OSI para aprovação de uma licença(não é necessário explicar cada um dos critérios)?

Os critérios da **OSI** para aprovação de uma licença de código aberto garantem que o software seja livre para uso, modificação e redistribuição, sem discriminação contra usuários ou áreas de atividade, e que o código-fonte esteja disponível para todos.

14. O que é terminal?

É uma interface de linha de comando (CLI - Command Line Interface) que permite aos usuários interagir com o sistema operacional através de comandos digitados em texto

15. Apresenta as principais vantagens e desvantagens dos sistemas operacionais Windows, MacOS e Linux.

O Windows é amplamente utilizado, com vasta compatibilidade de software e hardware, tornando-o ideal para usuários comuns e empresas, mas pode ser suscetível a vírus e falhas de desempenho. O macOS oferece uma interface intuitiva e é otimizado para hardware Apple, sendo popular entre designers e desenvolvedores, mas é limitado a dispositivos Apple e tem menos opções de software. O Linux é altamente customizável, seguro e gratuito, ideal para desenvolvedores e servidores, mas pode ser difícil para iniciantes e possui menos suporte para software comercial. Cada sistema tem suas vantagens, dependendo do uso e preferências do usuário.

16. Quem é o principal responsável pelas conexões internas de um computador?

O principal responsável pelas conexões internas de um computador é o chipset da placa mãe, pois atua como um intermediário, facilitando o fluxo de dados entre essas partes e garantindo que o computador funcione de maneira eficiente.

17. O que é o PID?

Ele é usado para distinguir e gerenciar processos no sistema, permitindo que o sistema operacional e os administradores de sistema controlem, monitorem e interajam com esses processos de forma eficiente.

18. O que é FHS?

É um padrão que define a estrutura e organização dos diretórios em sistemas operacionais baseados em Unix, como o Linux. Ele especifica onde os arquivos e diretórios devem ser localizados no sistema de arquivos, facilitando a consistência e a interoperabilidade entre diferentes distribuições do Linux.

19. O que é endereço IP?

É um identificador único atribuído a cada dispositivo conectado a uma rede

20. Qual a diferença entre o usuário root e o usuário comum?

O Usuário root tem total acesso ao sistema, com permissões completas, já o usuário comum possui permissões limitadas.

21. O que um candidato da certificação Linux Essentials deve saber?

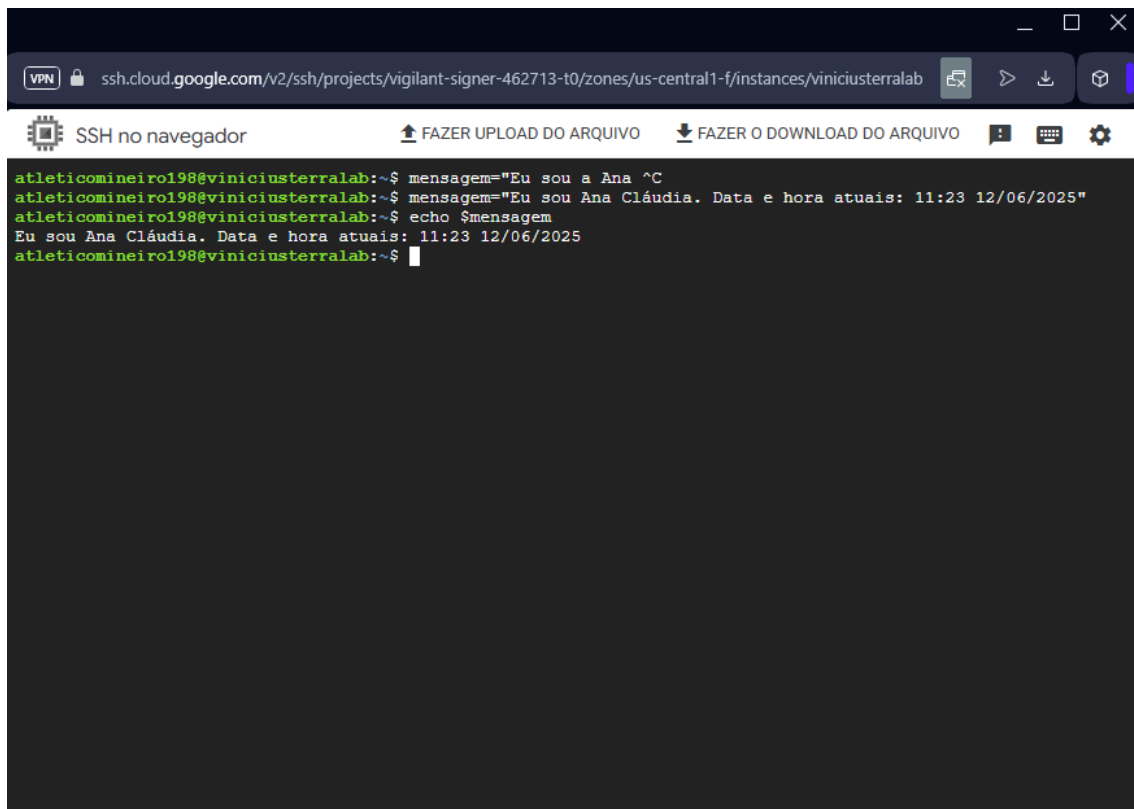
O candidato à certificação Linux Essentials deve ter uma boa compreensão do sistema operacional Linux, ser capaz de realizar tarefas básicas de administração, navegar no sistema de arquivos, gerenciar pacotes e entender conceitos fundamentais de rede e segurança.

### **Atividades Práticas utilizando o terminal**

#### **Instruções:**

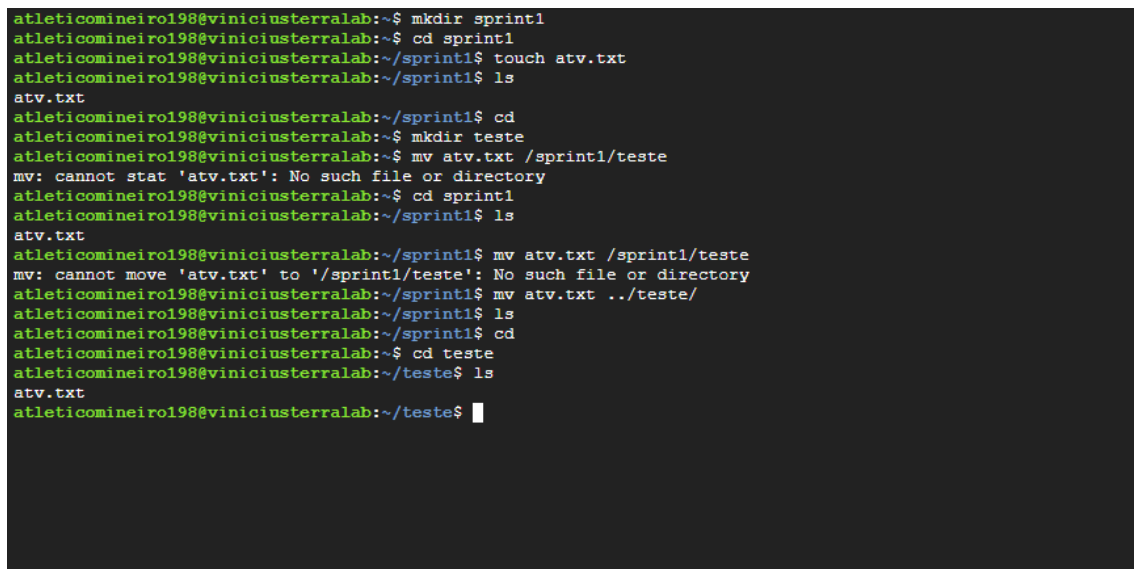
- **Todos os exercícios devem ser feitos utilizando linha de comando, não sendo permitido a utilização de qualquer interface gráfica que não seja o terminal;**
- **A resposta dos exercícios deve ser dada por meio de prints das execuções dos comandos necessários para chegar na resposta correta;**
- **Faça as observações que achar necessário;**

22. Crie uma variável com o texto “Eu sou Ana Cláudia. Data e hora atuais: x” sendo x a data e a hora no momento da execução. Mostre o conteúdo da variável criada no terminal. Obtenha o caminho do diretório atual.



```
atleticomineiro198@viniciusterralab:~$ mensagem="Eu sou a Ana ^C
atleticomineiro198@viniciusterralab:~$ mensagem="Eu sou Ana Cláudia. Data e hora atuais: 11:23 12/06/2025"
atleticomineiro198@viniciusterralab:~$ echo $mensagem
Eu sou Ana Cláudia. Data e hora atuais: 11:23 12/06/2025
atleticomineiro198@viniciusterralab:~$
```

23. Crie um diretório vazio. Dentro deste diretório crie um arquivo txt chamado “atv.txt”. Depois do arquivo criado, liste os arquivos presentes no diretório. Copie o arquivo “atv.txt” para outro diretório. Mova o arquivo “atv.txt para outro diretório”.



```
atleticomineiro198@viniciusterralab:~$ mkdir sprint1
atleticomineiro198@viniciusterralab:~$ cd sprint1
atleticomineiro198@viniciusterralab:~/sprint1$ touch atv.txt
atleticomineiro198@viniciusterralab:~/sprint1$ ls
atv.txt
atleticomineiro198@viniciusterralab:~/sprint1$ cd
atleticomineiro198@viniciusterralab:~$ mkdir teste
atleticomineiro198@viniciusterralab:~$ mv atv.txt /sprint1/teste
mv: cannot stat 'atv.txt': No such file or directory
atleticomineiro198@viniciusterralab:~$ cd sprint1
atleticomineiro198@viniciusterralab:~/sprint1$ ls
atv.txt
atleticomineiro198@viniciusterralab:~/sprint1$ mv atv.txt /sprint1/teste
mv: cannot move 'atv.txt' to '/sprint1/teste': No such file or directory
atleticomineiro198@viniciusterralab:~/sprint1$ mv atv.txt ../teste/
atleticomineiro198@viniciusterralab:~/sprint1$ ls
atv.txt
atleticomineiro198@viniciusterralab:~/sprint1$ cd
atleticomineiro198@viniciusterralab:~$ cd teste
atleticomineiro198@viniciusterralab:~/teste$ ls
atv.txt
atleticomineiro198@viniciusterralab:~/teste$
```

24. Acesse o manual de printf na seção número 3.

25. Criar um arquivo com o diretório criado anteriormente e mover o arquivo para fora do diretório. Em seguida, comprimir e descomprimir esse arquivo utilizando zip, gzip e bzip2.

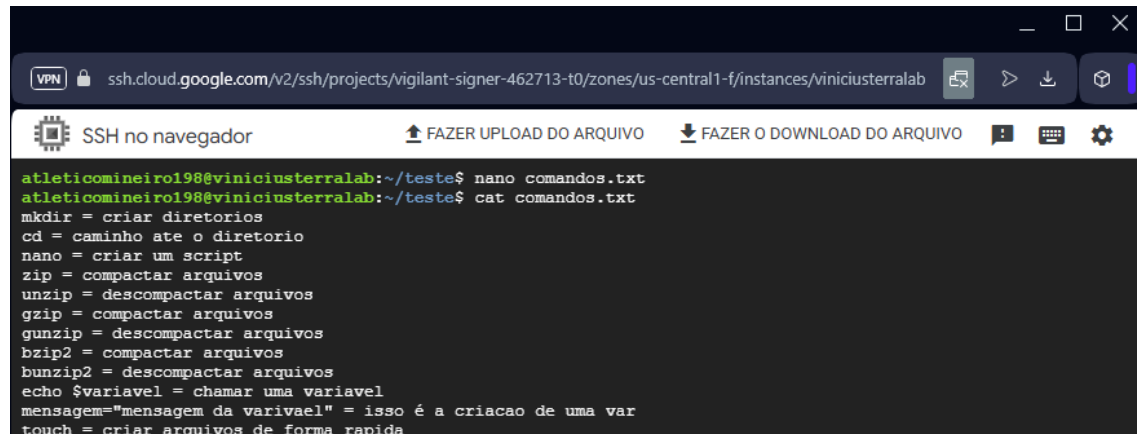
```
atleticomineiro198@viniusterrallab:~/sprint1$ touch readme.md
atleticomineiro198@viniusterrallab:~/sprint1$ ls
readme.md
atleticomineiro198@viniusterrallab:~/sprint1$ mv readme.md ../teste/
atleticomineiro198@viniusterrallab:~/sprint1$ ls ../teste/
atv.txt  readme.md
atleticomineiro198@viniusterrallab:~/sprint1$ cd
atleticomineiro198@viniusterrallab:~$ cd teste
atleticomineiro198@viniusterrallab:~/teste$ gzip readme.md
atleticomineiro198@viniusterrallab:~/teste$ bzip readme.md
-bash: bzip: command not found
atleticomineiro198@viniusterrallab:~/teste$ zip readme.zip readme.md
-bash: zip: command not found
atleticomineiro198@viniusterrallab:~/teste$ touch atividade.txt
atleticomineiro198@viniusterrallab:~/teste$ zip atividade.zip atividade.txt
-bash: zip: command not found
atleticomineiro198@viniusterrallab:~/teste$ sudo apt update && sudo apt install zip unzip
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:5 file:/etc/apt/mirrors/debian-security.list Mirrorlist [39 B]
Hit:7 https://packages.cloud.google.com/apt google-compute-engine-bookworm-stable InRelease
Hit:8 https://packages.cloud.google.com/apt cloud-sdk-bookworm InRelease
Hit:2 https://deb.debian.org/debian bookworm InRelease
Hit:3 https://deb.debian.org/debian bookworm-updates InRelease
Hit:9 https://packages.cloud.google.com/apt google-cloud-ops-agent-bookworm-2 InRelease
Hit:4 https://deb.debian.org/debian bookworm-backports InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
```

```
adding: atividade.txt (stored 0%)
atleticomineiro198@viniusterrallab:~/teste$ unzip atividade.zip
Archive:  atividade.zip
replace atividade.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
extracting: atividade.txt
atleticomineiro198@viniusterrallab:~/teste$ gzip atividade.txt
atleticomineiro198@viniusterrallab:~/teste$ gunzip arquivo.txt.gz
gzip: arquivo.txt.gz: No such file or directory
atleticomineiro198@viniusterrallab:~/teste$ ^C
atleticomineiro198@viniusterrallab:~/teste$ gunzip atividade.txt.gz
atleticomineiro198@viniusterrallab:~/teste$ bzip2 arquivo.txt
-bash: bzip2: command not found
atleticomineiro198@viniusterrallab:~/teste$ bzip2 atividade.txt
-bash: bzip2: command not found
atleticomineiro198@viniusterrallab:~/teste$ sudo apt update && sudo apt install bzip2
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:5 file:/etc/apt/mirrors/debian-security.list Mirrorlist [39 B]
Hit:7 https://packages.cloud.google.com/apt google-compute-engine-bookworm-stable InRelease
Hit:2 https://deb.debian.org/debian bookworm InRelease
Hit:3 https://deb.debian.org/debian bookworm-updates InRelease
Hit:8 https://packages.cloud.google.com/apt cloud-sdk-bookworm InRelease
Hit:4 https://deb.debian.org/debian bookworm-backports InRelease
Hit:9 https://packages.cloud.google.com/apt google-cloud-ops-agent-bookworm-2 InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
```



```
Processing triggers for man-db (2.11.2-2) ...
atleticomineiro198@viniciusterralab:~/teste$ bzip2 atividade.txt
atleticomineiro198@viniciusterralab:~/teste$ bunzip2 atividade.txt.bz2
atleticomineiro198@viniciusterralab:~/teste$
```

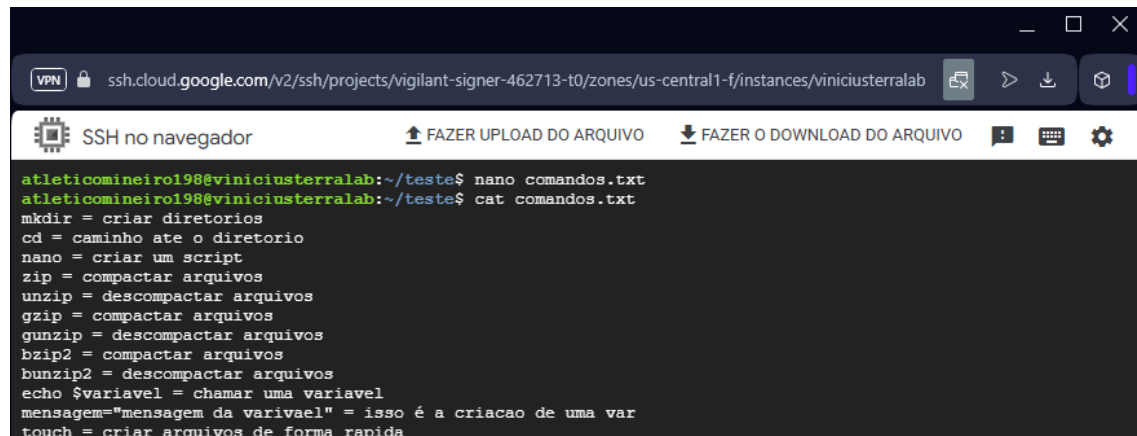
26. Criar e executar um script com todos os comandos apresentados no texto.



The screenshot shows a web browser window with the address bar displaying 'ssh.cloud.google.com/v2/ssh/projects/vigilant-signer-462713-t0/zones/us-central1-f/instances/viniciusterralab'. The browser interface includes a 'VPN' icon, a lock icon, and buttons for file upload and download. The terminal window shows the following commands and output:

```
atleticomineiro198@viniciusterralab:~/teste$ nano comandos.txt
atleticomineiro198@viniciusterralab:~/teste$ cat comandos.txt
mkdir = criar diretorios
cd = caminho ate o diretorio
nano = criar um script
zip = compactar arquivos
unzip = descompactar arquivos
gzip = compactar arquivos
gunzip = descompactar arquivos
bzip2 = compactar arquivos
bunzip2 = descompactar arquivos
echo $variavel = chamar uma variavel
mensagem="mensagem da variavel" = isso é a criação de uma var
touch = criar arquivos de forma rapida
```

27. Mostrar o conteúdo do arquivo "atv.txt".



The screenshot shows the same web browser window as the previous one. The terminal window shows the following commands and output:

```
atleticomineiro198@viniciusterralab:~/teste$ nano comandos.txt
atleticomineiro198@viniciusterralab:~/teste$ cat comandos.txt
mkdir = criar diretorios
cd = caminho ate o diretorio
nano = criar um script
zip = compactar arquivos
unzip = descompactar arquivos
gzip = compactar arquivos
gunzip = descompactar arquivos
bzip2 = compactar arquivos
bunzip2 = descompactar arquivos
echo $variavel = chamar uma variavel
mensagem="mensagem da variavel" = isso é a criação de uma var
touch = criar arquivos de forma rapida
```

28. Mostre os processos ativos de forma detalhada e por árvore genealógica.

```
Hit:7 https://packages.cloud.google.com/apt google-compute-engine-bookworm-stable InRelease
Hit:2 https://deb.debian.org/debian bookworm InRelease
Hit:3 https://deb.debian.org/debian bookworm-updates InRelease
atlaticomineiro198@viniciusterralab:~/teste$ pstree -p
systemd(1)
├─agetty(401)
├─agetty(402)
├─cron(838)
├─dbus-daemon(386)
├─exim4(703)
├─google_cloud_op(1689)
│   └─fluent-bit(1720)
│       ├──{fluent-bit}(1722)
│       ├──{fluent-bit}(1723)
│       ├──{fluent-bit}(1726)
│       ├──{fluent-bit}(1727)
│       ├──{fluent-bit}(1728)
│       ├──{fluent-bit}(1729)
│       ├──{fluent-bit}(1730)
│       ├──{fluent-bit}(1731)
│       ├──{fluent-bit}(1732)
│       ├──{fluent-bit}(1733)
│       ├──{fluent-bit}(1734)
│       ├──{fluent-bit}(1735)
│       ├──{fluent-bit}(1736)
│       ├──{fluent-bit}(1737)
│       ├──{fluent-bit}(1738)
│       ├──{fluent-bit}(1739)
│       ├──{fluent-bit}(1740)
│       ├──{fluent-bit}(1741)
│       ├──{fluent-bit}(1742)
│       ├──{fluent-bit}(1743)
│       └─{fluent-bit}(1744)
│   └─{google_cloud_op}(1691)
│       └─{google_cloud_op}(1692)
```

```
├─google_osconfig(415)
│   ├──{google_guest_co}(714)
│   ├──{google_osconfig}(678)
│   ├──{google_osconfig}(679)
│   ├──{google_osconfig}(680)
│   ├──{google_osconfig}(681)
│   ├──{google_osconfig}(803)
│   ├──{google_osconfig}(804)
│   └─{google_osconfig}(805)
├─haveged(379)
├─networkd-dispat(390)
├─otelopscol(1690)
│   ├──{otelopscol}(1696)
│   ├──{otelopscol}(1697)
│   ├──{otelopscol}(1698)
│   ├──{otelopscol}(1699)
│   ├──{otelopscol}(1700)
│   ├──{otelopscol}(1701)
│   └─{otelopscol}(1780)
├─polkitd(418)
│   ├──{polkitd}(463)
│   └─{polkitd}(467)
├─rsyslogd(394)
│   ├──{rsyslogd}(396)
│   ├──{rsyslogd}(397)
│   └─{rsyslogd}(399)
├─sshd(822)
│   ├──sshd(6723)
│   ├──sshd(6746)
│   ├──bash(6747)
│   └─pstree(10894)
├─systemd(6726)
│   ├──sd-pam(6727)
│   ├──systemd-journal(218)
│   ├──systemd-logind(831)
│   ├──systemd-network(258)
│   ├──systemd-resolve(370)
│   ├──systemd-timesyn(372)
│   ├──systemd-timesyn(375)
│   ├──systemd-udev(238)
│   └─unattended-upgr(419)
│       └─unattended-upgr(689)
```

29. Mostre o montante total de memória RAM, a quantidade de memória livre e o espaço de swap.

```
ssh.cloud.google.com/v2/ssh/projects/vigilant-signer-462713-t0/zones/us-central1-f/instances/viniciusterrallab
```

SSH no navegador

FAZER UPLOAD DO ARQUIVO FAZER O DOWNLOAD DO ARQUIVO

```
atleticomineiro198@viniciusterrallab:~/teste$ free -h
```

	total	used	free	shared	buff/cache	available
Mem:	3.8Gi	533Mi	2.5Gi	520Ki	1.1Gi	3.3Gi
Swap:	0B	0B	0B			

```
atleticomineiro198@viniciusterrallab:~/teste$
```

```
ssh.cloud.google.com/v2/ssh/projects/vigilant-signer-462713-t0/zones/us-central1-f/instances/viniciusterrallab
```

SSH no navegador

FAZER UPLOAD DO ARQUIVO FAZER O DOWNLOAD DO ARQUIVO

```
atleticomineiro198@viniciusterrallab:~$ vmstat -s
```

```
4018904 K total memory
570124 K used memory
322796 K active memory
905108 K inactive memory
2585208 K free memory
23036 K buffer memory
1079812 K swap cache
0 K total swap
0 K used swap
0 K free swap
16392 non-nice user cpu ticks
0 nice user cpu ticks
12083 system cpu ticks
1548220 idle cpu ticks
3084 IO-wait cpu ticks
0 IRQ cpu ticks
402 softirq cpu ticks
262 stolen cpu ticks
0 non-nice guest cpu ticks
0 nice guest cpu ticks
415416 K paged in
2039148 K paged out
0 pages swapped in
0 pages swapped out
8286176 interrupts
16276790 CPU context switches
1749736786 boot time
12959 forks
```

```
atleticomineiro198@viniciusterrallab:~$
```