4·Linux Open Software Specialists

Curso 707

PREPARATÓRIO RHCSA E LFCS

Sumário

1

Acessar um prompt shell e emitir comandos com sintaxe correta

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - **Acesse um prompt de shell e emita comandos com a sintaxe correta**
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Introdução.

O GNU/Linux faz uso de sua característica multi-usuário, ou seja, suporta vários usuários conectados ao mesmo tempo por meio dos **terminais virtuais**. Um terminal virtual é uma segunda seção de trabalho completamente independente de outras e pode ser acessado no computador local ou remotamente, a partir dos programas `telnet`, `rsh`, `rlogin`, `rdesktop`, `vnc`, `ssh`, etc. Nos dias de hoje, o acesso remoto é muito importante. A qualquer distância que se esteja do cliente, é possível atendê-lo.

No GNU/Linux é possível, em modo texto, acessar outros terminais virtuais segurando a tecla `ALT` e pressionando uma das teclas de F1 até F6. Cada tecla tem função correspondente a um número de terminal do `1` ao `6`. Esse é o comportamento padrão, embora possa ser mudado (o sétimo, por **default**, é usado pelo ambiente gráfico - **Xorg**).

O GNU/Linux possui mais de `63` terminais virtuais. Desses, apenas `6` estão disponíveis inicialmente por motivos de economia da memória "RAM". Se você estiver usando o modo gráfico, deve segurar `Ctrl + Alt` enquanto pressiona uma tecla de atalho de F1 a F6. Um exemplo prático: se você estiver utilizando o sistema no terminal `1`, pressione `Ctrl + Alt + F2`, e veja na primeira linha nome e versão do sistema operacional, nome da máquina e o terminal no qual você está.

Você pode utilizar quantos terminais quiser, do F1 ao F6 (inclusive utilizando o X) e pode ficar "saltando" de terminal para terminal.

Logon

Logon é a entrada do usuário, seja `root` ou usuario comum, onde deve ser digitado seu nome de usuário e logo depois sua senha. Caso você digite algo de forma errada, aparecerá uma mensagem de erro, impedinddo que você seja logado – autenticado - no sistema.

> É importante perceber que, quando se digita a senha, não aparece nenhum retorno, como os famosos asteriscos. O objetivo é evitar que um observador mais curioso seja capaz de contar quantos caracteres sua senha possui.

Shell

No Mundo `GNU/Linux`, utilizamos o `Shell`, que funciona como interpretador de comandos. Ele é a interface entre o usuário e o kernel do sistema e, por meio dele, podemos digitar os comandos. O `Shell` padrão do GNU/Linux é o `Bash`. Entretanto, existem também outras interfaces, como, por exemplo, `csh`, `tcsh`, `ksh` e `zsh`.

O kernel é a parte mais próxima do hardware do computador. É o núcleo do Sistema Operacional. Se seu GNU/Linux estiver com problemas, não chute seu computador, a culpa não é dele!

O local onde o comando será digitado é marcado por um **traço piscante** na tela, chamado de **cursor**. Tanto em Shell texto como em Shell gráfico é necessário o uso do **cursor** para saber onde devemos iniciar a digitação de textos e nos orientarmos quanto à posição na tela. Popularmente conhecido como linha de comando, o Shell interpreta a ação do usuário através das instruções digitadas.

Essas instruções poderão ser executadas por dois níveis de usuários, com permissões diferentes. São eles: - **Super usuário** - Popularmente conhecido como **root**. O usuário **root** é o administrador do sistema, e seu diretório (pasta) padrão é o /root, diferentemente dos demais usuários que ficam dentro do /home. O Shell de um usuário **root** se diferencia do Shell de um usuário comum, pois antes do cursor ele é identificado com # (jogo-da-velha). - **Usuário comum** - É qualquer usuário do sistema que não seja **root** e não tenha poderes administrativos no sistema. Como já havíamos dito anteriormente, o diretório padrão para os usuários é o /home. Antes do cursor, o Shell de um usuário comum é identificado com $ (cifrão).

Existem muitas funcionalidades no Shell, uma delas é retornar comandos que já foram digitados anteriormente. Para fazer isso, é só pressionar as teclas seta para cima e seta para baixo para ter acesso ao histórico de comandos. Inclusive o nome do programa responsável por manter essa lista é history.

Outra funcionalidade muito utilizada é de rolar a nossa tela de modo que possamos ir para cima ou para baixo, parecido com o scroll. Para rolarmos a tela para cima, segura-se a tecla Shift e em seguida pressione o Page Up. Para rolarmos a tela para baixo, segura-se a tecla Shift e pressionamos o Page Down.

Isto é útil para ver textos que rolaram rapidamente para cima e saíram do nosso campo de visão. A execução de comandos com poderes administrativos exige que o nível do usuário comum seja alterado. Uma das formas de fazer isso é utilizando o comando su - Super User . Veja sua descrição abaixo: - **su** - Para usar o comando su, é necessário ter o password do administrador. Uma vez que o nível tenha sido mudado será possível executar qualquer comado com poderes de root. Após logar com usuário aluno, utilize o comando su:

```
$ su
```

Será pedido a senha do usuário root. Após efetuar a autenticação do usuário, o prompt mudará de $ para #, indicando que você está logado como administrador do sistema.

Existem dois comandos, `whoami` e `who am i`, que permitem saber quem você é em determinado momento. A sequência de comandos abaixo esclarece o uso e finalidade destes dois comandos claramente:

```
[root@localhost 4linux]# whoami
[root@localhost 4linux]# who am i
```

O comando `whoami` indica quem você é no momento **root**. Se você utilizou o comando `su` para tornar-se outro usuário, o comando `whoami` informa quem você realmente é - `aluno`, pois foi com ele que você logou na máquina antes de trocar de usuário.

Ele também pode ser utilizado para trocar de usuário. Não pedirá a senha se você for usuário root:

```
[4linux@localhost ~]$ su - aluno
```

Com a opção `-`, além de trocar o user, também carregará as variáveis locais do usuário:

```
[4linux@localhost ~]$ su -
```

Histórico de comandos

O terminal do `GNU/Linux` permite que você guarde `500` comandos por padrão no Debian e `1000` comandos no CentOS.

```
[root@localhost 4linux]# history
```

FC

`FC` significa **Find Command** ou **Fix Command**, pois ele executa as duas tarefas, encontra e corrige comandos. Para listar os comandos já digitados, guardados no history, digite:

```
[root@localhost 4linux]# fc -l
```

Por padrão, aparecem os últimos 16 comandos. Para visualizar uma lista de comandos do 2 ao 6, faça:

```
[root@localhost 4linux]# fc -l 2 6
```

Para visualizar os últimos 20 comandos:

```
[root@localhost 4linux]# fc -l -20
```

Para visualizar todos os comandos desde o último, começando com h:

```
[root@localhost 4linux]# fc -l h
```

Logout

Logout é a saída do sistema. Ela é feita por um dos comandos abaixo:

```
[root@localhost 4linux]# logout
[root@localhost 4linux]# exit
[root@localhost 4linux]# <CTRL >+D
```

Ou quando o sistema é reiniciado ou desligado.

Desligando o Computador

Para desligar o computador, é possível utilizar um dos comandos abaixo sempre quese esteja com o nível de usuário **root**:

```
[root@localhost 4linux]# shutdown -h now
[root@localhost 4linux]# halt
[root@localhost 4linux]# poweroff
```

Acessando os diretórios

Vamos aprender agora alguns comandos essenciais para a nossa movimentação dentro do sistema. O comando `pwd` exibe o diretório corrente. Ele é muito útil quando estamos navegando pelo sistema e não lembramos qual é o diretório atual.

```
[root@localhost 4linux]# pwd
```

O comando `cd` é utilizado para mudar o diretório atual, onde o usuário está. Ir para o diretório `home` do usuário logado:

```
[root@localhost 4linux]# cd
[root@localhost 4linux]# cd ~
```

Ir para o início da árvore de diretórios, ou seja, o diretório `/`:

```
[root@localhost 4linux]# cd /
```

Ir para um diretório específico:

```
[root@localhost 4linux]# cd /etc
```

Sobe um nível na árvore de diretórios:

```
[root@localhost 4linux]# cd ..
```

Retorna ao diretório anterior:

```
[root@localhost 4linux]# cd -
```

Entra em um diretório específico:

```
```

```
[root@localhost 4linux]# cd /usr/share/doc
```

Sobe 2 níveis da árvore de diretórios:

```
[root@localhost 4linux]# cd  ../../
```

> Atenção! Note a diferença entre caminhos absolutos e relativos: - **Absolutos**: `/etc/ppp`; `/usr/share/doc`; `/lib/modules` - **Relativos**: `etc/ppp`; `../doc`; `../../usr`;

Diretório . e ..

> Fique esperto para conhecer as diferenças entre o `.` e o `..` e o que eles representam para o sistema. Os comandos de movimentação muitas vezes são grandes alvos nas provas, por isso uma boa interpretação pode ser necessária, já que você pode precisar deles para resolver uma questão maior.

O comando `ls` é utilizado para listar o conteúdo dos diretórios. Se não for especificado nenhum diretório, ele mostrará o conteúdo daquele onde estamos no momento. Liste o conteúdo do diretório atual:

```
[root@localhost 4linux]# ls
```

Com os atalhos do bash a seguir, vamos testar algumas funcionalidades da linha de comando. Não é necessário se preocupar em decorá-los, com o passar do tempo, pegamos um pouco mais de prática: - Pressione a tecla **Back Space** para apagar um caractere à esquerda do cursor; - Pressione a tecla **Delete** para apagar o caractere corrente no cursor; - Pressione a tecla **Home** para ir ao começo da linha de comando; - Pressione a tecla **End** para ir ao final da linha de comando; - Pressione as teclas **Ctrl + A** para mover o cursor para o início da linha de comandos; - Pressione as teclas **Ctrl + E** para mover o cursor para o fim da linha de comandos; - Pressione as teclas **Ctrl + U** para apagar o que estiver à esquerda do cursor. O conteúdo apagado é copiado e pode ser colado com **Ctrl + y**; - Pressione as teclas **Ctrl + K** para apagar o que estiver à direita do cursor. O conteúdo apagado é copiado e pode ser colado com **Ctrl + y**; - Pressione as teclas **Ctrl + l** para limpar a tela e manter a linha de comando na primeira linha. Se você der um **Shift + Page Up**, ainda consegue enxergar o conteúdo. O **Ctrl + l** é um atalho para o comando **clear**; - Pressione as tecla **Ctrl + c** para

abrir uma nova linha de comando, na posição atual do cursor; - Pressione as teclas **Ctrl + d** para sair do **Shell**. Este é equivalente ao comando **exit**; - Pressione as teclas **Ctrl + r** para procurar **x** letra relacionada ao último comando digitado que tinha **x** letra como conteúdo do comando. - Executar o último comando pressione:**!!** - Executar um comando específico do histórico de comandos:`!<numero>`, ou seja,`!12`.

Usar redirecionamento de entrada-saída (>, », |, 2>, etc.)

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - **Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)**
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Introdução

No mundo GNU/Linux, a maioria das operações são realizadas por meio de comandos escritos. Em geral, eles permitem um maior controle e flexibilidade de operações, além de poderem ser incluídos em `scripts`. Nesta aula, aprenderemos alguns comandos avançados.

Trabalhando com entrada e saída de dados

No linux, você pode ler dados de um arquivo ou terminal ou escrever dados para um arquivo ou terminal. O linux tem três tipos de fluxo de dados: - Entrada(`INPUT`) - Saída(`OUTPUT`) - e a última para imprimir diagnósticos ou mensagens de erro.

Por padrão, a entrada de dados e comandos no `Shell` é feita pelo teclado, a saída destes é retornada na tela. Eventuais erros são exibidos na tela também. Porém você pode alterar a saída padrão que é a tela e enviá-la para um arquivo ou outra localização.

Os termos geralmente usados são: - **0** - Entrada de dados, representada por `stdin` (Standard Input; - **1** - Saída de dados, representada por `stdout`(Standard Output); - **2** - Saída de erros, representada por `stderr`(Standard Error);

Alterando os redirecionamentos

Formas de redirecionar o fluxo de dados: - > - (maior) - Direciona a saída do comando para um arquivo, substituindo seu conteúdo, caso o arquivo já exista; - **ż** - (maior-maior) - Direciona a saída do comando para um arquivo, adicionando o texto ao final do arquivo, caso ele já exista; - < - (menor) - Passa o conteúdo do arquivo como argumento para o comando; - **2**> - (dois-maior) - Direciona as saídas de erro geradas pelo programa para um arquivo,substituindo seu conteúdo, caso o arquivo já exista; - **2ż** - (dois-maior-maior) - Direciona as saídas de erro geradas pelo programa para um arquivo, adicionando o texto ao final do arquivo, caso ele já exista; - **2>&1** - (dois-maior-e-um) - Direciona as saídas de erro para a saída do comando, no caso para `STDOUT`; - **&**> - (e-maior) - Direciona todas as saídas (normal e de erro) para um arquivo, substituindo seu conteúdo, caso ele já exista; - **&ż** - (e-maior-maior) - Direciona todas as saídas (normal e de erro) para um arquivo, adicionando o texto ao final do arquivo, caso ele já exista; - | - (barra vertical ou pipe) - Utiliza a saída do primeiro comando como argumento do segundo comando; - **tee** - Mostra saída na tela e redireciona para um arquivo ou outra localização ao mesmo tempo; - **ń** - Marca o fim de um bloco.

O direcionador >

O direcionador > direciona a saída padrão de um comando para um arquivo. Caso o arquivo exista, seu conteúdo é substituído. Vejamos, então, uma saída do comando `ls`:

```
[root@localhost 4linux]# ls /
```

Para gravar essa lista em um arquivo chamado `raiz`, utilizamos o direcionador da seguinte forma:

```
[root@localhost 4linux]# ls / > raiz
```

Não aparece nada na tela porque o comando foi executado sem erros e sua saída redirecionada para o arquivo `raiz`, confira:

```
[root@localhost 4linux]# cat raiz
```

O conteúdo do arquivo `raiz` é o mesmo da saída do comando `ls`. Cuidado ao utilizar o direcionador para o mesmo arquivo, pois os dados serão perdidos, exemplo:

Quero enviar a saída do arquivo raiz para raiz2: - 1 - Primeiro, visualize o arquivo para ver que há dados no arquivo `raiz`:

```
[root@localhost 4linux]# cat raiz
```

- 2 - Depois, envie a saída do `cat` para o arquivo `raiz2`:

```
[root@localhost 4linux]# cat raiz > raiz
```

Ao realizar o comando acima, a primeira interpretação do bash é executar o comando: "> **raiz2**", ou seja, se não existe o arquivo, ele será criado, e se já existe é sobrescrito.

No caso, ele sobrescreve o arquivo raiz, deixando-o em branco. Assim, quando o comando `cat raiz` é executado, não há saída, pois o arquivo está zerado, não redirecionando nada.

Para evitar esse problema, execute o comando:

```
[root@localhost 4linux]# set -o noclobber
```

Após o comando, faça o exemplo:

```
[root@localhost 4linux]# cat /etc/fstab  > hoje
```

```
[root@localhost 4linux]# cat  hoje > hoje
```

Verifique que o arquivo não foi sobrescrito e para voltar:

```
[root@localhost 4linux]# set +o noclobber
```

O direcionador >>

O direcionador `>>` direciona a saída padrão de um comando para um arquivo. Caso o arquivo exista, a saída é adicionada ao final do arquivo.

```
[root@localhost 4linux]# ls / >> hoje
```

Verifique que a saída do comando `ls` foi adicionada ao final do arquivo `hoje`.

O direcionador <

O direcionador `<` é utilizado para passar um `stdin` para um comando, ele é geralmente utilizado para passar o conteúdo de arquivos como parâmetros de comandos.

Alguns comandos precisam que seja passado o `stdin` para eles serem executados, vamos ver o exemplo do comando `tr`, que traduz ou deleta caracteres:

Para converter letras minúsculas em maiúsculas, faça:

```
[root@localhost 4linux]# tr "a-z" "A-Z" /etc/passwd
```

Verifique que sem o redirecionador `<` o comando não é executado com sucesso, agora faça corretamente:

```
[root@localhost 4linux]# tr "a-z" "A-Z" < /etc/passwd
```

Você também pode utilizar o comando `tr` para deletar caracteres. Vamos deletar as vogais do arquivo:

```
[root@localhost 4linux]# tr -d aeiou  < /etc/passwd
```

Para que as mudanças sejam efetuadas de fato é necessário encaminhar a saída para outro arquivo.

O direcionador 2>

Quando utilizamos o direcionador `>`, ele não redireciona as saída de erro, apenas a saída sem erros. Caso o arquivo não exista será criado e caso já exista será sobrescrito.

Por exemplo, vamos usar o comando `ls` usando como parâmetro um diretório que não existe e redirecionar sua saída para um novo arquivo:

```
[root@localhost 4linux]# ls nao_existe > ls_naoexiste
ls: impossível  acessar  nao_existe: Arquivo  ou  diretório nãoencontrado
```

Verifique que mesmo não redirecionando a saída com erro, o arquivo `ls_naoexiste` é criado:

```
[root@localhost 4linux]# cat ls_naoexiste
```

Para gravar as mensagens de erro, devemos utilizar o direcionador `2>`:

```
[root@localhost 4linux]# ls nao_existe 2> ls_naoexiste.err
```

Agora sim, nenhuma mensagem de erro foi exibida na tela, porque ela foi enviada para o arquivo `ls_naoexiste.err`. Vamos verificar o conteúdo dele:

```
[root@localhost 4linux]# cat ls_naoexiste.err
ls: impossível acessar nao_existe: Arquivo ou diretório nãoencontrado
```

O direcionador 2»

Quando utilizamos o direcionador `2»`, ele redireciona apenas as mensagens de erro. Caso o arquivo não exista será criado e caso já exista será adicionada a saída ao final do arquivo.

```
[root@localhost 4linux]# cat ls_naoexiste.err
```

Agora, vamos redirecionar outra saída de erro para este arquivo:

```
[root@localhost 4linux]# cat /nada 2>> ls_naoexiste.err
```

Verifique que a saída de erro foi adicionada ao arquivo `ls_naoexiste.err`:

```
[root@localhost 4linux]# cat ls_naoexiste.err2
ls: impossível acessar nao_existe: Arquivo ou  diretório não encontrado
cat: /nada: Arquivo ou diretório não encontrado
```

O direcionador 2>&1

Podemos usar os direcionadores `>` e `2>`, em conjunto, para gerar um arquivo com a saída padrão e outro com a saída de erros, desta forma:

```
[root@localhost 4linux]# cat /etc/*
```

A saída mostra tanto o conteúdo dos arquivos quanto os erros por tentar ler um diretório com o comando `cat`. Vamos enviar a saída deste comando para arquivos diferentes:

```
[root@localhost 4linux]# cat /etc/* > msg_correto 2> msg_errado
```

Visualize o conteúdo dos arquivos `msg_correto` e `msg_errado`:

```
[root@localhost 4linux]# cat msg_correto
[root@localhost 4linux]# cat msg_errado
```

Mas e se for necessário gravar todas as mensagens em um arquivo apenas? Nesse caso, podemos redirecionar o `stderr` para o `stdout`:

```
[root@localhost 4linux]# cat /etc/* > msg_total  2>&1
```

Aqui, redirecionamos o `stdout` para o arquivo `msg_total` e redirecionamos o `stderr` para `stdout`, ou seja, também para o arquivo `msg_total`. Visualize seu conteúdo:

```
[root@localhost 4linux]# cat msg_total
```

O direcionador &>

Podemos usar os direcionadores `>` e `2>` em conjunto, para gerar um arquivo com a saída padrão e outro com a saída de erros, desta forma:

```
[root@localhost 4linux]# cat /etc/*
```

A saída mostra tanto o conteúdo dos arquivos quanto os erros por tentar ler um diretório com o comando `cat`. Vamos enviar a saída deste comando para arquivos diferentes:

```
[root@localhost 4linux]# cat /etc/* > msg_ok 2> msg_error
```

Visualize o conteúdo dos arquivos `msg_ok` e `msg_error`:

```
[root@localhost 4linux]# cat msg_ok
[root@localhost 4linux]# cat msg_error
```

Mas e se for necessário gravar todas as mensagens em um arquivo apenas? Para isso existe o direcionador `&>`, que direciona tanto as mensagens padrão quanto as mensagens de erro para

um único arquivo. Caso o arquivo não exista, será criado; mas se já existir, será sobrescrito. Repetindo o teste anterior:

```
[root@localhost 4linux]# cat /etc/* &> ls_out
```

Não aparece nenhuma mensagem no terminal, pois tanto as mensagens ok quanto as mensagens com erro foram redirecionadas para o arquivo `ls_out`, visualize seu conteúdo:

```
[root@localhost 4linux]# cat ls_out
```

O direcionador &»

Assim como o redirecionador `&>`, redireciona tanto a saída de `stdout` quanto a saída de `stderr` para um único arquivo, a diferença é que, caso o arquivo não exista, ele será criado. Caso já exista, será adicionado a saída com comando ao final do arquivo. Visualize o arquivo `ls_out`:

```
[root@localhost 4linux]# cat ls_out
```

Agora redirecione a saída `stdout` e `stderr` para ele com `&»`:

```
[root@localhost 4linux]# cat /etc/* &>> ls_out
```

Não aparece nenhuma mensagem no terminal, pois tanto as mensagens ok quanto as mensagens com erro foram redirecionadas para o arquivo `ls_out`, visualize seu conteúdo:

```
[root@localhost 4linux]# cat ls_out
```

Observe que a saída foi adicionada ao final do arquivo.

O direcionador |

Conhecido como `pipe`, ele envia o `stdout` de um comando para o `stdin` do próximo comando para dar continuidade ao processamento. Os dados enviados serão processados pelo próximo

comando, trazendo assim um resultado esperado. Vamos usar novamente o comando `tr` para exemplificar, mas desta vez utilizando o `pipe`:

Primeiro, visualize o conteúdo do arquivo `/etc/passwd`:

```
[root@localhost 4linux]# cat /etc/passwd
```

A saída foi o `stdout` do comando.

Vamos agora redirecionar este `stdout` para o comando `tr`:

```
[root@localhost 4linux]# cat /etc/passwd | tr "a-z" "A-Z"
```

O direcionador `tee`

Quando usado junto com o pipe `|`, o `tee` permite que a saída padrão do comando seja exibida na tela e enviada para um arquivo ao mesmo tempo. Veja a saída de um comando e envie-a para um arquivo qualquer, caso o arquivo não exista, será criado e caso já exista será sobrescrito, caso queira adicionar à um arquivo já existente use `tee -a` :

```
[root@localhost 4linux]# cat /etc/fstab | tee arquivo.tee
```

A saída aparece na tela e também foi direcionada para o arquivo `arquivo.tee`, visualize-o:

```
[root@localhost 4linux]# cat arquivo.tee
```

O direcionador «

Temos ainda o direcionador «, utilizado para marcar o fim de exibição de um bloco. Um dos usos mais frequentes desse direcionador é em conjunto com o comando `cat`. Você pode editar um novo arquivo com o comando `cat` ou até mesmo adicionar conteúdo nele, veja:

```
[root@localhost 4linux]# cat << EOF > arquivo_novo
```

> Onde: «EOF - indica que a edição do arquivo terminará quando em uma linha contiver apenas a sequência EOF.

- > arquivo_novo - direciona o que for digitado no arquivo para `arquivo_novo`. Ex:

```
[root@localhost 4linux]# cat << EOF > arquivo_novo
Este
é
meu arquivo!
EOF
```

Visualize o arquivo gerado:

```
[root@localhost 4linux]# cat arquivo_novo
```

2

Usar grep e expressões regulares para analisar o texto

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - **Use grep e expressões regulares para analisar o texto**
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Introdução

O comando `grep` é um dos comandos mais úteis em um ambiente de terminal Linux. O nome `grep` significa "impressão de expressão regular global". Isso significa que você pode usar `grep` para verificar se a entrada que ele recebe corresponde a um padrão especificado. Esse comando, além de ser trivial, é extremamente poderoso: sua capacidade de classificar a entrada com base em regras complexas o torna um link popular em muitas cadeias de comando.

`Grep` pode ser usado para combinar padrões literais em um arquivo de texto. Isso significa que se você passar ao `grep` uma palavra para pesquisar, ele imprimirá todas as linhas do arquivo que contém essa palavra. Uma expressão regular, regex ou regexp, é uma sequência de caracteres que define um padrão de pesquisa. Normalmente, esses padrões são usados por algoritmos de pesquisa de string para operações **localizar** ou **localizar e substituir**, ou para validação de entrada.

> Nesta aula, usaremos o `grep` para pesquisar várias palavras e frases na documentação do CUPS - Servidor de Impressões do Linux. Acompanhe no video como obteremos tal arquivo.

Busca literal

Na forma mais básica, você usa o comando `grep` para combinar padrões literais em um arquivo de texto. Isso significa que se você passar uma palavra para o `grep` pesquisar, todas as linhas do arquivo que contém essa palavra serão impressas.

Neste exemplo, o primeiro argumento `CUPS` é o padrão que estamos procurando, enquanto o segundo argumento `CUPS` é o arquivo de entrada que desejamos pesquisar.

```
[root@localhost 4linux]# grep "CUPS" CUPS
```

Pesquisa que não diferencia maiúsculas de minúsculas

Por padrão, `grep` simplesmente pesquisará o padrão exato, especificado no arquivo de entrada e retornando as linhas que encontrar. Podemos tornar esse comportamento mais útil adicionando algumas flags opcionais ao `grep`. Se quisermos que o `grep` ignore o "caso" de nosso parâmetro de pesquisa e procure variações em maiúsculas e minúsculas, podemos especificar a opção `-i` ou `--ignore-case`. Pesquisaremos cada instância da palavra `licence` (com maiúsculas, minúsculas ou maiúsculas e minúsculas) no mesmo arquivo de antes.

```
[root@localhost 4linux]# grep -i "license" CUPS
```

Inverter pesquisa

Se quisermos encontrar todas as linhas que **não** contenham um padrão especificado, podemos usar a opção `-v` ou `--invert-match`. Podemos pesquisar todas as linhas que não contenham a palavra `the` na licença BSD com o seguinte comando:

```
[root@localhost 4linux]# grep -iv "cups" CUPS
```

Muitas vezes é útil saber o número da linha em que ocorrem as correspondências. Você pode fazer isso usando a opção `-n` ou `--line-number`. Execute novamente o exemplo anterior com esta flag:

```
[root@localhost 4linux]# grep -vn "CUPS" CUPS
```

Âncoras

Âncoras são caracteres especiais que especificam onde na linha uma correspondência deve ocorrer para ser válida. Por exemplo, usando âncoras, podemos especificar que queremos apenas saber sobre as linhas que correspondem a `CUPS` no início da linha. Para fazer isso, podemos usar a âncora `^` antes da string literal. Este exemplo de string somente corresponderá se `CUPS` ocorrer no início de uma linha.

```
[root@localhost 4linux]# grep -i "^CUPS" CUPS
```

Da mesma forma, a âncora `$` pode ser usada após uma string para indicar que a correspondência só será válida se ocorrer no final de uma linha. Vamos corresponder todas as linhas que terminam com a palavra ponto . (com escape `\.`) na seguinte expressão regular:

```
[root@localhost 4linux]# grep -i "\.$" CUPS
```

Combinando com qualquer caractere

O caractere ponto (.) é usado em expressões regulares para significar que qualquer caractere único pode existir no local especificado. Por exemplo, se quisermos corresponder a qualquer coisa que tenha dois caracteres e a string `PS`, podemos usar o seguinte padrão:

```
[root@localhost 4linux]# grep -i "..PS" CUPS
```

Expressões de colchetes

Colocando um grupo de caracteres entre colchetes, podemos especificar que o caractere naquela posição pode ser qualquer caractere encontrado dentro do grupo de colchetes. Isso significa que, se quiséssemos encontrar as linhas que contêm `too` ou `two`, poderíamos especificar essas variações sucintamente usando o seguinte padrão:

```
[root@localhost 4linux]# grep "t[wo]o" CUPS
```

A notação de colchetes também nos permite algumas opções interessantes. Podemos fazer com que o padrão corresponda a qualquer coisa, exceto aos caracteres entre colchetes, iniciando a lista de caracteres dentro dos colchetes com um caractere `^`.

```
[root@localhost 4linux]# grep -i "[^c]ode" CUPS
```

Outro recurso útil dos colchetes é que você pode especificar um intervalo de caracteres em vez de digitar individualmente cada caractere disponível. Isso significa que se quisermos encontrar cada linha que começa com uma letra maiúscula, podemos usar o seguinte padrão:

```
[root@localhost 4linux]# grep "^[A-Z]" CUPS
```

ou

```
[root@localhost 4linux]# grep "^[[:upper:]]" CUPS
```

Repitindo o padrão zero ou mais vezes

Um dos metacaracteres mais comumente usados é o asterisco, ou `*`, que significa "repita o caractere ou expressão anterior zero ou mais vezes".

Para localizar cada linha no arquivo `CUPS` que contém um parêntese de abertura e fechamento, com apenas letras e espaços simples entre eles, use a seguinte expressão:

```
[root@localhost 4linux]# grep "([A-Za-z ]*)" CUPS
```

Encontrando uma Palavra em um arquivo de Texto

Para pesquisar uma palavra em um arquivo de texto basta executar o comando:

```
grep solicitação arquivo
```

- solicitação – A palavra que você está pesquisando.
- arquivo – O arquivo em que a busca está sendo realizada.

Em nosso caso, estamos procurando pela palavra `joatham` em um arquivo chamado `4linux`:

```
grep joatham /tmp/4linux
```

Contador de Palavras

Com o comando `grep` você pode descobrir quantas vezes a palavra pesquisada aparece no arquivo de texto. Basta adicionar a opção `-c`.

```
grep -c solicitação arquivo
```

Pesquisando por Múltiplas Palavras

Até agora mostramos apenas exemplos em que pesquisamos uma única palavra. O `grep` também suporta pesquisas de várias palavras em um único comando, ficando assim:

```
grep solicitação1 arquivo | grep solicitação2 arquivo
```

O comando trabalha de maneira simples. Primeiro, pesquisamos pela Solicitação1, então o comando vai para a próxima palavra – Solicitação2.

Encontrando uma Palavra entre Vários Arquivos

Também é possível pesquisar em diversos arquivos por uma palavra em um único comando:

```
grep -l solicitação ./*
```

3

Acessando máquinas remotas usando SSH

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - **Acesse sistemas remotos usando SSH**
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Introdução

Há uma variedade de programas que fornecem essa funcionalidade aos desenvolvedores. `Telnet` é um programa de terminal que permite aos usuários fazer login em computadores remotos na mesma rede. `Ftp` fornece uma maneira de transferir arquivos entre computadores. Eles já existem há muito tempo, mas sua fraqueza está na falta de segurança.

`SSH`, ou Secure Socket Shell, fornece aos usuários acesso seguro a máquinas remotas por meio de uma conexão criptografada. Todos os dados enviados do cliente (seu computador) são criptografados. Somente quando os dados chegarem ao servidor remoto ele será descriptografado. O processo de criptografia de dados é transparente. Isso acontece nos bastidores e não interrompe seu fluxo de trabalho, este é um grande benefício. Ferramentas como `telnet` e `ftp` não fornecem essa camada de segurança.

Acesse sistemas remotos usando SSH

O SSH permite que você acesse computadores e/ou servidores remotamente, desde que o serviço esteja instalado e funcionando. O uso básico do comando é o seguinte:

```
[root@localhost 4linux]# ssh user@hostname
```

O nome do host pode ser um endereço IP ou um nome de domínio acessível a partir do computador do qual você está tentando fazer o ssh. A porta padrão para SSH é `22`, mas você também pode ter seu serviço `sshd` ouvindo em uma porta diferente se você configurou para fazer isso. Se o seu servidor `SSH` estiver escutando na porta `2222`, poderíamos facilmente fazer o login definindo a porta em nosso comando `ssh`.

```
[root@localhost 4linux]# ssh -p 2222 user@hostname
```

Outro sinalizador útil que você pode usar é o sinalizador `-X`, que habilita o encaminhamento do X11, permitindo que você execute programas GUI na rede em sua máquina local.

```
[root@localhost 4linux]# ssh -X user@hostname
```

Chaves SSH

Uma das principais vantagens de usar SSH são as chaves SSH. Trata-se de um par de chaves criptográficas públicas/privadas usadas para autenticação, que sempre vêm em pares. A chave privada é armazenada no cliente, já a chave pública é armazenada na máquina remota. Quando um usuário tenta se conectar a uma máquina remota via SSH, o protocolo verifica o computador do usuário em busca da chave privada que corresponde à chave pública armazenada na máquina remota. Se houver uma correspondência, a conexão foi bem-sucedida. Nenhuma senha necessária! Você pode até adicionar uma senha para aumentar a segurança.

Como configurar chaves SSH

Essas etapas o ajudarão a configurar as chaves SSH em seu próprio servidor privado. Todos os comandos serão inseridos por meio do terminal. Para isso, abra o terminal em sua máquina local e execute o seguinte comando:

```
[root@localhost 4linux]# ssh -keygen -t rsa
```

A opção `-t` especifica que tipo de chave gerar. Especificamos uma chave com o tipo `rsa`. É apenas um tipo de chave baseado no algoritmo RSA. Existem várias opções que você pode adicionar ao comando acima. Observe que duas chaves foram geradas: a primeira é sua chave privada, que será armazenada em seu computador local. A segunda é a chave pública. Deve ser uma oferta inoperante por causa da extensão `.pub`. Esta é a chave que será colocada na máquina remota.

A chave pública deve ser colocada em qualquer máquina remota que você planeja acessar via SSH. digite:

```
[root@localhost 4linux]# ssh-copy-id -i ~/.ssh/<public> <user>@<remote>
```

Depois de executar o comando, você verá algumas saídas que podem parecer alarmantes à primeira vista. Não se preocupe. É apenas porque esta é a primeira vez que você está tentando autenticar com a máquina remota por meio de chaves SSH. O comando `ssh-copy-id` move o arquivo de chave pública especificado com `-i ~/.ssh/<public>`do seu computador para `<remote>`. Ele será armazenado no arquivo `~/.ssh/authorized_keys`; no diretório inicial de `<user>`.

Vá em frente e tente fazer o login na máquina remota usando:

```
[root@localhost 4linux]# ssh <user>@<remote>
```

Você notará que efetuou login na máquina remota sem digitar uma senha. Sucesso! Você está autenticado!# Fazer login e alternar usuários em destinos com vários usuários

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - **Faça login e troque de usuários em destinos multiusuário**
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Introdução

Como a maioria dos administradores de sistema, quando a gente pensa no primeiro programa no `init` e no `SystemV`, automaticamente, a gente pensa na inicialização e desligamento do Linux e basicamente como gerenciar serviços uma vez que estejam funcionando. Como o `init`, o `systemd` é a mãe de todos os processos, sendo responsável por trazer o host Linux a um estado em que o trabalho seja bem mais produtivo. Algumas das funções assumidas pelo `systemd`, que são muito mais extensa do que o antigo programa `init`, servem para gerenciar muitos aspectos de um host Linux em execução, incluindo montagem de sistemas de arquivos, gerenciamento de hardware, manipulação de temporizadores e inicialização e gerenciamento de serviços de sistema necessários para ter um host Linux produtivo.

Portanto, o `Systemd` é um conjunto de ferramentas que fornece um modelo de inicialização rápido e flexível para gerenciar uma máquina inteira desde sua inicialização. Ele fornece um gerenciador de sistema e serviço que funciona como `PID 1` e controla a inicialização do resto do sistema. Nos últimos anos, a maioria das distribuições Linux adotou `systemd` como sistema

`init` padrão.

Inicialização do Linux

O processo completo que leva um host Linux de um estado desligado para um estado de execução é complexo, mas é aberto e pode ser conhecido. Antes de entrar em detalhes, daremos uma visão geral de quando o hardware do host é ligado até que o sistema esteja pronto para um usuário fazer login. Na maioria das vezes, **o processo de inicialização** é discutido como uma entidade única, mas isso não é exato. Existem, na verdade, três partes principais no processo de inicialização e inicialização completa:

- Inicialização de **hardware**: inicializa o hardware do sistema.
- **Inicialização do Linux**: carrega o kernel do Linux e, em seguida, o systemd.
- **Inicialização do Linux**: onde o systemd prepara o host para o trabalho produtivo.

A sequência de inicialização do Linux começa depois que o kernel carrega o `init` ou o `systemd` , dependendo se a distribuição usa a inicialização antiga ou nova, respectivamente. Os programas `init` e `systemd` iniciam e gerenciam todos os outros processos e são conhecidos como a **mãe de todos os processos** em seus respectivos sistemas.

É importante separar a inicialização do hardware da inicialização do Linux e definir explicitamente os pontos de demarcação entre eles. Entender essas diferenças e que papel cada uma desempenha para levar um sistema Linux a um estado em que possa ser produtivo torna possível gerenciar esses processos e determinar melhor onde um problema está ocorrendo durante o que a maioria das pessoas chama de "inicialização".

O processo de inicialização segue o processo de inicialização de três etapas e traz o computador Linux a um estado operacional no qual é utilizável para trabalho produtivo. O processo de inicialização começa quando o kernel transfere o controle do host para o **systemd**.

Quais são os Targets que o `systemd` utiliza por padrão?

Enquanto você não criar o próprio serviço de boot — não se preocupe, ensinamos como no próximo tópico —, os runlevels do systemd serão carregados a partir das unidades Target padrões. Em outras palavras, unidades pré-configuradas que, em tese, englobam todos os serviços que o usuário comum necessita. São eles:

- **poweroff.target** - Desliga o sistema e o computador (power off);
- **rescue.target** - Aciona o modo rescue pelo shell;
- **multi-user.target** - Configura o sistema para multiusuários sem interface gráfica;
- **graphical.target** - Usa um sistema multiusuário com interface gráfica e serviços de

rede; e
- **reboot.target** - Desliga e reinicia o sistema.

Os systemd runlevels não são marcados por números, e sim por nomes acompanhados da extensão .target.

No entanto, é evidente a equivalência deles aos do `systemvinit`, de modo que o `multi-user.target` tem a mesma função dos níveis `2`, `3` e `4`, enquanto o `graphical.target` tem a função do runlevel `5` do `sysvinit`.

Para efeito de experimento, abra o terminal e digite o comando `systemctl --type=service`. Veja que é aberta uma imensa (provavelmente) lista de serviços em execução, contendo o status e a descrição de cada um. Isso nada mais são que unidades de serviço presentes no `graphical.target`.

Editando os Runlevels com o `systemd`

Modificar `systemd runlevels` é uma tarefa ainda mais fácil que a anterior. Mas por onde começar? Acho uma boa ideia descobrirmos qual é o runlevel que utilizamos por padrão. Digite:

```
[root@localhost 4linux]# systemctl get-default

graphical.target
```

Traduzindo, o primeiro comando solicitou ao systemd que informasse o `Target` padrão (default.target). Aliás, a menos que o usuário faça modificações, o `default.target` sempre será esse na maioria dos casos.

Retomando o exemplo do tópico anterior, é interessante que façamos a troca do runlevel padrão para o `multi-user.target`, pois ele executará o script que criamos há pouco. Faça isso da seguinte forma:

```
[root@localhost 4linux]# systemctl set-default multi-user.target
```

Muito simples, não é mesmo? Outra possibilidade bastante útil é a alternância de runlevels com o sistema operacional em execução. O resultado do procedimento é que somente os serviços que fazem parte do Target passarão a funcionar. Exemplo:

```
[root@localhost 4linux]# systemctl isolate multi-user.target
```

Fará com que o systemd execute o runlevel multi-user.target

```
[root@localhost 4linux]# systemctl isolate graphical.target
```

Retorna ao runlevel graphical.target

Alternar usuários com o comando su

O comando su é usado para executar uma função como um usuário diferente. É a maneira mais fácil de mudar ou mudar para a conta administrativa na sessão de login atual.

Algumas versões do Linux desabilitam a conta do usuário root por padrão, tornando o sistema mais seguro. No entanto, isso também restringe o usuário de executar comandos específicos.

Usar su para atuar temporariamente como usuário root permite que você ignore essa restrição e execute tarefas diferentes com usuários diferentes.

> Uma conta root é uma conta de administrador mestre com acesso total e permissões no sistema. Devido à gravidade das alterações que essa conta pode fazer e ao risco de ser comprometida, a maioria das versões do Linux usa contas de usuário limitadas para uso normal.

Sintaxe do comando su

Para usar o comando su, insira-o em uma linha de comando da seguinte maneira:

```
su [options] [username [arguments]]
```

Se um nome de usuário for especificado, o padrão su é o superusuário (root). Basta encontrar o usuário de que você precisa e adicioná-lo à sintaxe do comando su.

Opções de comando su

Para exibir uma lista de comandos, digite o seguinte:

```
[4linux@localhost ~]$ su -h
```

Aqui estão algumas opções comuns para usar com o comando `su`:

- **Username** – Substitui o nome de usuário pelo nome de usuário real com o qual deseja fazer login. Pode ser qualquer usuário, não apenas root.
- **-c ou –command [command]** - Executa um comando específico com o usuário especificado. –command [command]
- - ou **-l** ou **–login [username]** – Executa um script de login para alterar para um nome de usuário específico. Você precisará inserir uma senha para esse usuário.
- **-s** ou **–shell shell** – Permite que você especifique um ambiente de shell diferente para execução.
- **-h** ou **–help** – Mostrar o arquivo de ajuda para o comando `su`.
- **-p** ou **–preserve–environment** – Preserve o ambiente do shell (HOME, SHELL, USER, LOGNAME).

Exemplos de comando su

Mudar para um usuário diferente Para trocar o usuário conectado nesta janela de terminal, digite o seguinte:

```
[4linux@localhost ~]$ su -l [other_user]
```

Você será solicitado a fornecer uma senha. Digite e o login será alterado para esse usuário.

Se você omitir um nome de usuário, o padrão será a conta `root`. Agora, o usuário conectado pode executar todos os comandos do sistema. Isso também mudará o diretório inicial e o caminho para os arquivos executáveis.

Use o comando `whoami` para verificar se você mudou para um usuário diferente.

Executar um comando específico como um usuário diferente

Para executar um comando específico como um usuário diferente, use a opção `-c`:

```
[4linux@localhost ~]$ su -c [command] [other_user]
```

O sistema responderá solicitando a senha do usuário.

Ao inserir este exemplo, o sistema usará a conta especificada para executar o comando `ls` (listar o conteúdo do diretório).

Use um `Shell` diferente

Para usar um shell ou ambiente operacional diferente, digite o seguinte:

```
[4linux@localhost ~]$ su -s /usr/bin/zsh
```

Este comando abre uma conta de usuário root no **Z shell**.

Use um usuário diferente no mesmo ambiente

Você pode manter o ambiente da conta do usuário atual com a opção `-p`:

```
[4linux@localhost ~]$ su -p [other_user]
```

Substitua [other_user] pelo nome de usuário real para o qual deseja alternar.

A conta do usuário mudará, mas você manterá o mesmo diretório inicial. Isso é útil se para executar um comando como um usuário diferente, mas precisa de acesso aos dados do usuário atual.

Para verificar se você permaneceu no mesmo ambiente doméstico, use o comando `echo $HOME`, que exibirá o diretório em que está trabalhando.

4

Arquivar, compactar, desempacotar e descomprimir arquivos .tar, .star, .gzip e .bzip2

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - **Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2**
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

O empacotador `tar`

A compressão e empacotamento de arquivos e diretórios é muito importante em qualquer sistema computacional. Ambos os procedimentos são necessários para distribuição de softwares, economia de banda e espaço de armazenamento e backup do sistema.

O programa `tar`, cujo nome deriva de `tape archiver`, realiza a tarefa de concatenar todos os arquivos e diretórios, preservando as informações do arquivosystem, isto é, seus metadados. Criado com propósito de backup em dispositivos de acesso sequencial (unidadesde fita), o `tar` é utilizado, hoje em dia, como uma ferramenta de empacotamento, podendo ser utilizado em conjunto com compactadores como `gzip` ou `bzip2`.

A utilização da ferramenta `tar` é bastante simples. Seguindo a filosofia Unix, faça apenas uma tarefa, mas faça bem feito. O `tar` é um programa especialista em empacotar vários arquivos. Dessa forma, quando utilizamos os parâmetros `z` ou `j`, estamos na realidade fazendo uma chamada externa aos comandos `gzip` ou `bzip2`, especialistas em compressão de dados. Outros programas que trabalham de forma análoga ao `tar` são o `dump` e `cpio`. Ambos foram criados com a mesma finalidade, mas são pouco utilizados hoje em dia, pois não são tão versáteis quanto o `tar`.

Compactadores `GZIP`, `BZIP2`

Compressão de dados é o processo de codificar a informação de forma que seja possível armazená-la em um número menor de **bits**. Por exemplo, se definíssemos que a palavra **compressão** passaria a ser abreviada por **comp**, estaríamos diminuindo o número de **bits** necessários para armazenar esta apostila. Entretanto, para que você pudesse entender o que **comp** significa, seria necessário estar ciente dessa convenção - ou seja, do algoritmo de compressão. Há dois tipos básicos de compressão, aquele em que não há perdas de informações e aquele em que elas ocorrem. Obviamente, quando o assunto é **backup** de informações vitais, devemos utilizar algoritmos sem perdas.

Já em arquivos de imagens, vídeos e áudio, há casos em que podemos nos dar ao luxo de perdas de informações em detrimento da qualidade, que em geral é praticamente imperceptível para os não especialistas da área. Os principais programas de compressão que utilizaremos são o `bzip2` e `gzip`. O `bzip2` utiliza os algoritmos *Burrows-Wheeler transform* e *Huffman coding*; já o `gzip` utiliza os algoritmos *LZ77* e *Huffman coding*. Todos esses algoritmos fazem parte do grupo dos algoritmos que não ocasionam perdas de dados. A forma de utilização desses comandos é bastante simples.

Para o `gzip`, `bzip2`, basta fornecer o arquivo de entrada que a compressão se dará no próprio arquivo. Eis uma diferença entre o `tar` e esses programas: ele recebe dois argumentos, os arquivos de entrada e o arquivo de saída, ou seja, aqueles a serem empacotados e comprimidos.

Verifique que não é possível compactar um diretório sem empacotá-lo antes.

Hands On

Antes de usar o `tar`, existem algumas flags com as quais você precisa se familiarizar.

- **-c** - Criar um novo arquivo.
- **-p** - Manter as permissões originais do(s) arquivo(s);
- **-r** - Acrescenter arquivos a um arquivo tar;
- **-t** - Exibir o conteúdo de um arquivo tar;
- **-v** - Exibir detalhes da operação;
- **-x** - Extrair arquivos de um arquivo tar;
- **-z** - Comprimir ou extrair arquivos tar resultante com o gzip;
- **-j** - Comprimir ou extrair arquivos tar resultante com o bzip2;
- **-f** - Especificar o arquivo tar a ser usado;
- **-C** - Trocar o diretório para local de armazenamento ou restauração de dados

Montando a estrutura

```
[root@localhost 4linux]# mkdir -p /tmp/diretorio_teste/subdiretorio1
[root@localhost 4linux]# mkdir -p /tmp/diretorio_teste/subdiretorio2
[root@localhost 4linux]# mkdir -p /tmp/diretorio_teste/subdiretorio3
[root@localhost 4linux]# mkdir -p /tmp/extrair-aqui
[root@localhost 4linux]# touch /tmp/diretorio_teste/subdiretorio1/arquivo1.txt
[root@localhost 4linux]# touch /tmp/diretorio_teste/subdiretorio1/arquivo2.txt
[root@localhost 4linux]# touch /tmp/diretorio_teste/subdiretorio2/arquivo3.txt
[root@localhost 4linux]# touch /tmp/diretorio_teste/subdiretorio2/arquivo4.txt
[root@localhost 4linux]# touch /tmp/diretorio_teste/subdiretorio3/arquivo5.txt
[root@localhost 4linux]# touch /tmp/diretorio_teste/subdiretorio3/arquivo6.txt
```

Comando `tar`

Crie um arquivo:

```
[root@localhost 4linux]# cd /tmp
[root@localhost 4linux]# tar -cvf arquivo1.tar diretorio_teste
```

Verifique o conteúdo:

```
[root@localhost 4linux]# tar -tvf /tmp/arquivo1.tar
```

Extraia:

```
[root@localhost 4linux]# cd /tmp/extrair-aqui
[root@localhost 4linux]# tar -xvf /tmp/arquivo1.tar
```

Comando `star`

O comando `star` pode não ser instalado por padrão, mas você pode instalá-lo com o seguinte comando:

```
[root@localhost 4linux]# yum install star
```

Crie um arquivo:

```
[root@localhost 4linux]# cd /tmp
[root@localhost 4linux]# star -cv f=arquivo2.star diretorio_teste
```

Verifique o conteúdo:

```
[root@localhost 4linux]# star -tv f=/tmp/arquivo2.star
```

Extraia:

```
[root@localhost 4linux]# cd /tmp/extrair-aqui
[root@localhost 4linux]# star -xv f=/tmp/arquivo2.star
```

Comando `gzip`

O comando `gzip` compacta os arquivos especificados, dando a eles uma extensão `.gz`. Neste caso, usaremos para compactar um arquivo `.tar`.

```
[root@localhost 4linux]# cd /tmp
[root@localhost 4linux]# tar -cvf arquivo3.tar diretorio_teste
[root@localhost 4linux]# gzip arquivo3.tar
```

A opção `-z` do comando `tar` permite que você faça isso diretamente.

```
[root@localhost 4linux]# cd /tmp
[root@localhost 4linux]# tar -cvzf arquivo3.tar.gz diretorio_teste
```

Os arquivos são descompactados usando o comando `gunzip`.

```
[root@localhost 4linux]# gunzip arquivo3.tar.gz
```

A opção `-z` do comando `tar` permite que você descompacte e extraia diretamente um arquivo `.tar.gz`.

```
[root@localhost 4linux]# cd /tmp/extrair-aqui
[root@localhost 4linux]# tar -xvzf /tmp/arquivo3.tar.gz
```

Comando `bzip2`

O comando `bzip2` é semelhante ao comando `gzip`. Ele compacta os arquivos especificados, dando-lhes uma extensão `.bz2`. Neste caso, usaremos para compactar um arquivo `.tar`.

```
[root@localhost 4linux]# cd /tmp
[root@localhost 4linux]# tar -cvf arquivo4.tar diretorio_teste
[root@localhost 4linux]# bzip2 arquivo4.tar
```

A opção `-j` do comando `tar` permite que você faça isso diretamente.

```
[root@localhost 4linux]# cd /tmp
[root@localhost 4linux]# tar -cvjf arquivo4.tar.bz2 diretorio_teste
```

Os arquivos são descompactados usando o comando `bunzip2`.

```
[root@localhost 4linux]# bunzip2 arquivo4.tar.bz2
```

A opção `-j` do comando `tar` permite que você extraia diretamente um arquivo `.tar.bz2`.

```
shell [root@localhost 4linux]# cd /tmp/extrair-aqui [root@localhost 4linux]# tar -xvjf /tmp/arquivo4
.tar.bz2# Criar e editar arquivos de texto
```

Pontos de estudo para o exame

Os candidatos ao exame `RHCSA` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - **Crie e edite arquivos de texto**
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Introdução

Usar um editor de texto para criar ou editar seus próprios arquivos é uma tarefa tão óbvia que todo sistema operacional sempre tem um editor de texto pré-instalado. Você pode ter que editar arquivos em seu computador, independentemente de serem arquivos de configuração, arquivos html de seus sites ou, mais comumente, arquivos de texto simples.

Entre os vários editores de texto do mundo Linux, o `Vim` (ou Vi IMproved) destaca-se pela versatilidade e pelas funções que oferece. Na verdade, o Vim é capaz de acelerar a escrita do código, fornecendo alguns atalhos para realizar todas as operações de modificação, exclusão ou substituição do texto.

O Vim Text Editor também permite que você instale diferentes plug-ins através dos quais transformar este editor de texto simples em um IDE real para programação em diferentes linguagens.

Inicialmente, o usuário, ao se aproximar desta ferramenta pela primeira vez, pode se sentir

confuso com todos os comandos disponíveis. No entanto, depois de memorizar os principais comandos de edição, você poderá dispensar essa ferramenta!

Embora mais engenhoso, o `vim` é totalmente compatível com versões anteriores do `vi`, tornando ambos indistinguíveis para a maioria das tarefas.

A maneira padrão de iniciar o `vi` é fornecer a ele um caminho para um arquivo como parâmetro. Para pular diretamente para uma linha específica, seu número deve ser informado com um sinal de mais, como em `vim +9 /etc/fstab` para abrir `/etc/fstab/` e posicionar o cursor na `9ª` linha. Sem um número, o sinal de mais por si só coloca o cursor na última linha.

Modo de inserção

O modo de inserção é simples: o texto aparece na tela à medida que é digitado no teclado. É o tipo de interação que a maioria dos usuários espera de um editor de texto, mas não é como o `vim` primeiro apresenta um documento. Para entrar no modo de inserção, o usuário deve executar um comando de inserção no modo normal. A tecla `Esc` termina o modo de inserção e retorna ao modo normal, o modo `vim` padrão.

> Se você estiver interessado em saber mais sobre os outros modos de execução, abra o vi e digite:

> :help vim-modes-intro

Modo normal

O modo normal, também conhecido como modo de comando, é como o `vim` inicia por padrão.

Neste modo, o teclado é associado a comandos para navegação e tarefas de manipulação de texto. A maioria dos comandos neste modo são chaves exclusivas. Algumas das teclas e suas funções no modo normal são:

- **0, $**: vá para o início e o fim da linha;
- **1G, G**: vá para o início e o final do documento;
- **(,)**: vá para o início e o final da frase;
- **{,}**: vá para o início e o final do parágrafo;
- **w, W**: palavra de salto e palavra de salto incluindo pontuação;
- **h, j, k, l**: esquerda, baixo, cima, direita;

- **e ou E**: vá para o final da palavra atual;
- **/,?**: pesquise para a frente e para trás;
- **i, I**: entre no modo de inserção antes da posição atual do cursor e no início da linha atual;
- **a, A**: entre no modo de inserção após a posição atual do cursor e no final da linha atual;
- **o, O**: adicione uma nova linha e entre no modo de inserção na próxima linha ou na linha anterior;
- **s, S**: apague o caractere sob o cursor ou a linha inteira e entre no modo de inserção;
- **c**: altere os caracteres sob o cursor;
- **r**: substitua o caractere sob o cursor;
- **x**: exclua os caracteres selecionados ou o caractere sob o cursor;
- **v, V**: inicie uma nova seleção com o caractere atual ou a linha inteira;
- **y, yy**: copie (puxa) o (s) personagem (s) ou a linha inteira;
- **p, P**: cole o conteúdo copiado, antes ou depois da posição atual;
- **u**: desfaça a última ação;
- **Ctrl-R**: refaça a última ação;
- **ZZ**: feche e salve;
- **ZQ**: feche e não salve.

Se precedido por um número, o comando será executado o mesmo número de vezes. Por exemplo, pressione `3yy` para copiar a linha atual mais as duas seguintes, pressione `d5w` para excluir a palavra atual e as 4 palavras seguintes, e assim por diante.

A maioria das tarefas de edição são combinações de vários comandos. Por exemplo, a sequência de teclas `vey` é usada para copiar uma seleção começando na posição atual até o final da palavra atual. A repetição de comandos também pode ser usada em combinações, então `v3ey` copiaria uma seleção começando na posição atual até o final da terceira palavra a partir daí.

O `vim` pode organizar o texto copiado em registros, permitindo manter conteúdos distintos ao mesmo tempo. Um registro é especificado por um caractere precedido por “e, uma vez criado, é mantido até o final da sessão atual. A sequência de teclas"ly cria um registro contendo a seleção atual, que será acessível por meio da tecla l. Então, o registro l pode ser colado com "lp.

Também existe uma maneira de definir marcas personalizadas em posições arbitrárias ao longo do texto, tornando mais fácil alternar rapidamente entre elas. As marcas são criadas pressionando a tecla **m** e, em seguida, uma tecla para endereçar a posição atual. Feito isso, o cursor voltará para a posição marcada quando a tecla escolhida for pressionada.

Comandos de dois pontos

O modo normal também oferece suporte a outro conjunto de comandos do vi: os comandos de dois pontos. Os comandos de dois pontos, como o nome indica, são executados após pressionar a tecla de dois pontos : no modo normal. Os comandos de dois pontos permitem ao usuário realizar pesquisas, salvar, sair, executar comandos do shell, alterar as configurações do vi etc. Para voltar ao modo normal, o comando **:visual** deve ser executado ou a tecla Enter pressionada sem nenhum comando. Alguns dos comandos de dois pontos mais comuns são indicados aqui (a inicial não faz parte do comando):

- **:s/REGEX/TEXT/g**: Substitua todas as ocorrências da expressão regular REGEX por TEXT na linha atual. Ele aceita a mesma sintaxe do comando sed, incluindo endereços;
- **:!**: Execute o seguinte comando shell;
- **:quit ou :q**: Saia do programa;
- **:quit! ou :q!**: Saia do programa sem salvar;
- **:wq**: Salvar e sair;
- **:exit ou :x ou :e**: Salve e saia, se necessário;
- **:visual**: Volte para o modo de navegação.

O programa **vim** padrão é capaz de fazer a maioria das tarefas de edição de texto, mas qualquer outro editor não gráfico pode ser usado para editar arquivos de texto no ambiente shell.

> Usuários iniciantes podem ter dificuldade em memorizar as teclas de comando do vi de uma só vez.

> As distribuições que adotam o vim também possuem o comando **vimtutor**, que usa o próprio `vim` para abrir um guia passo a passo das principais atividades.

> O arquivo é uma cópia editável que pode ser usada para praticar os comandos e progressivamente se acostumar com eles.

> Talvez isso possa te ajudar... **Fonte:** https://medium.com/usevim/vim-cheat-sheet-poster-2745f2b2162c # Criar, excluir, copiar e mover arquivos e diretórios ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - **Crie, exclua, copie e mova arquivos e diretórios**
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Introdução

Se você não sabia, no Linux tudo são arquivos, então é importantíssimo saber como manipulá-los. Em geral, um usuário Linux precisa saber navegar pelo sistema de arquivos, copiar arquivos de um local para outro e excluir arquivos.

Bom, mas o que é um arquivo? Um arquivo é uma entidade que armazena dados e programas. Consiste em conteúdo e metadados (tamanho do arquivo, proprietário, data de criação, permissões etc). Os arquivos são organizados em diretórios. Um diretório é um arquivo que armazena outros arquivos.

Dentre os diferentes tipos de arquivos, temos:

- **Arquivos regulares**: armazenam dados e programas.
- **Diretórios**: contêm outros arquivos.
- **Arquivos especiais**: usados para entrada e saída de dados.

Claro, também existem outros tipos de arquivos além desses principais.

Criar, copiar, mover e remover arquivos

Criando arquivos com touch

O comando **touch** é a maneira mais fácil de criar arquivos novos e vazios. Também pode ser usado para alterar os carimbos de data/hora (ou seja, hora de modificação) dos arquivos e diretórios existentes. A sintaxe para usar touch é:

```
touch OPTIONS FILE_NAME(S)
```

Sem nenhuma opção, **touch** cria novos arquivos com os nomes de arquivo fornecidos como argumentos, desde que ainda não existam arquivos com o mesmo nome. Além disso, o comando **touch** pode criar qualquer número de arquivos simultaneamente:

```
[root@localhost 4linux]# touch arquivo1 arquivo2 arquivo3
```

Com esse comando, são criados três arquivos novos vazios: arquivo1, arquivo2 e arquivo3.

Diversas opções de **touch** foram especificamente pensadas para permitir ao usuário alterar os carimbos de data/hora dos arquivos. Por exemplo, a opção -a altera apenas a hora de acesso, enquanto a opção **-m** altera apenas a hora de modificação. O uso de ambas as opções em conjunto altera ambas as horas de acesso e de modificação para o horário atual:

```
[root@localhost 4linux]# touch -am arquivo3
```

Copiando arquivos com cp

Como usuários Linux, geralmente copiamos arquivos de um local para outro. Podemos usar o cp para todas as tarefas de cópia, seja para mover um arquivo de música ou um arquivo de sistema de um diretório para outro:

```
[root@localhost 4linux]# cp arquivo1 dir2/
```

Este comando pode ser interpretado literalmente como copiar arquivo1 para o diretório `dir2`. O resultado é a presença de arquivo1 dentro de `dir2`. Para que este comando seja executado com sucesso, arquivo1 deve existir no diretório atual do usuário. Caso contrário, o sistema exibe a mensagem de erro **No such file or directory**:

```
[root@localhost 4linux]# cp dir1/arquivo1 dir2
cp: cannot stat 'dir1/arquivo1': No such file or directory
```

Neste caso, observe que o caminho para arquivo1 é mais explícito. O caminho de origem pode

ser expresso como um caminho relativo ou um caminho absoluto. Os caminhos relativos são dados em referência a um diretório específico, ao passo que os caminhos absolutos não são fornecidos com uma referência. Esclareceremos melhor essa noção mais adiante.

Por enquanto, apenas observe que este comando copia arquivo1 para o diretório dir2. O caminho para arquivo1 é fornecido com mais detalhes, pois o usuário atualmente não está localizado em dir1:

```
[root@localhost 4linux]# cp /home/4linux/arquivo2 /home/4linux/Documents/
```

Neste terceiro caso, arquivo2, localizado em **/home/4linux/4linux**, é copiado no diretório **/home/4linux/Documents/**. O caminho fornecido aqui é absoluto. Nos dois exemplos acima, os caminhos são relativos. Quando um caminho começa com o caractere `/`, trata-se de um caminho absoluto; caso contrário, ele é relativo.

A sintaxe geral de `cp` é:

```
cp OPTIONS SOURCE DESTINATION
```

SOURCE é o arquivo a ser copiado e DESTINATION o diretório para o qual o arquivo será copiado. SOURCE e DESTINATION podem ser especificados como caminhos absolutos ou relativos.

Movendo arquivos com o mv

Assim como o **cp** para copiar, o Linux fornece um comando para mover e renomear arquivos. Ele se chama **mv**.

A operação de mover é análoga à de recortar e colar que costumamos executar por meio de uma interface gráfica de usuário (GUI).

Se quiser mover um arquivo para um novo local, use mv da seguinte maneira:

```
mv FILENAME DESTINATION_DIRECTORY
```

Aqui está um exemplo:

```
[root@localhost 4linux]# mv arquivo1 /home/debian/Documents/
```

O resultado é que arquivo1 é movido para o destino /home/debian/Documents.

Para renomear um arquivo, **mv** é usado da seguinte maneira:

```
[root@localhost 4linux]# mv arquivo1 4linuxfile
```

Esse comando muda o nome do arquivo de arquivo1 para 4linuxfile.

Por padrão, o mv não pede confirmação se você quiser sobrescrever (renomear) um arquivo existente. No entanto, podemos fazer com que o sistema exiba uma mensagem, usando a opção **-i**:

```
[root@localhost 4linux]# mv -i 4linuxagain teste
mv: replace 'teste', overriding mode 0644 (rw-r--r--)?
```

Este comando solicita a permissão do usuário antes de sobrescrever 4linuxagain com teste.

Inversamente, se usarmos **-f**, o arquivo será sobrescrito à força, sem pedir permissão:

```
[root@localhost 4linux]# mv -f teste 4linuxdenovo
```

Removendo arquivos com rm

rm é usado para excluir arquivos. Pense nele como uma forma abreviada da palavra "remover". Note que a ação de remover um arquivo geralmente é irreversível e, portanto, este comando deve ser usado com cautela:

```
[root@localhost 4linux]# rm 4linuxdenovo
```

Este comando excluiria 4linuxdenovo:

```
[root@localhost 4linux]# rm -i 4linuxdenovo
rm: remove write-protected regular empty file '4linuxdenovo'?
```

Este comando solicitaria uma confirmação ao usuário antes de excluir 4linuxdenovo. Lembre-se, vimos a opção **-i** ao usarmos **mv**, acima:

```
[root@localhost 4linux]# rm -f 4linuxdenovo
```

Este comando exclui 4linuxdenovo à força, sem pedir sua confirmação.

Podemos excluir vários arquivos ao mesmo tempo:

```
[root@localhost 4linux]# rm arquivo1 arquivo2 arquivo3
```

Neste exemplo, arquivo1, arquivo2 e arquivo3 são excluídos simultaneamente.

A sintaxe de **rm** geralmente é a seguinte:

```
rm OPTIONS FILE
```

Criando e removendo diretórios

Criando diretórios com mkdir

A criação de diretórios é essencial para organizar seus arquivos e pastas. Os arquivos podem ser agrupados de maneira lógica dentro de um diretório. Para criar um diretório, use **mkdir**:

```
mkdir OPTIONS DIRECTORY_NAME
```

onde DIRECTORY_NAME é o nome do diretório a ser criado. Podemos criar qualquer número de diretórios simultaneamente:

```
[root@localhost 4linux]# mkdir dir1
```

criaria o diretório dir1 no diretório atual do usuário:

```
[root@localhost 4linux]# mkdir dir1 dir2 dir3
```

O comando anterior cria três diretórios, dir1, dir2 e dir3, ao mesmo tempo.

Para criar um diretório junto com seus subdiretórios, use a opção **-p** ("parents"):

```
[root@localhost 4linux]# mkdir -p joatham/pedro
```

Este comando criaria a estrutura de diretórios joatham/pedro, ou seja, criaria os diretórios joatham e pedro. O diretório pedro estaria localizado dentro do diretório joatham.

Removendo diretórios com o rmdir

rmdir deleta um diretório se ele estiver vazio. Sua sintaxe é dada por:

```
rmdir OPTIONS DIRECTORY
```

Onde DIRECTORY pode ser um único argumento ou uma lista de argumentos:

```
[root@localhost 4linux]# rmdir dir1/
```

Este comando excluiria dir1.

```
[root@localhost 4linux]# rmdir dir2/ dir3/
```

Este comando excluiria simultaneamente dir1 e dir2.

Podemos remover um diretório junto com seu subdiretório:

```
[root@localhost 4linux]# rmdir -p joatham/pedro/
```

Isso removeria a estrutura de diretórios joatham/pedro. Note que, se algum dos diretórios não estiver vazio, ele não será excluído.

Manipulação recursiva de arquivos e diretórios

Para manipular um diretório e seu conteúdo, precisamos aplicar a recursão. Recursão significa efetuar uma ação e repetí-la em toda a árvore de diretórios. No Linux, as opções **-r** ou **-R** ou **–recursive** são geralmente associadas à recursão.

Cópia recursiva com `cp -r`

`cp -r` (ou -R ou –recursive) permite copiar um diretório junto com todos os seus subdiretórios e arquivos.

```
[root@localhost 4linux]# tree 4linux
4linux
 |--- dir2
 |      `-- arquivo1
 |--- arquivo2
  `-- arquivo3

1 directory, 3 files

[root@localhost 4linux]# mkdir novacopia
[root@localhost 4linux]# cp 4linux novacopia/
cp: -r not specified; omitting directory '4linux'
[root@localhost 4linux]# cp -r 4linux novacopia/
[root@localhost 4linux]# tree novacopia/
newcopy/
  `-- 4linux
     |--- dir2
     |      `-- arquivo1
     |--- arquivo2
      `-- arquivo3

2 directories, 3 files
```

Remoção recursiva com `rm -r`

`rm -r` remove um diretório e todo o seu conteúdo (subdiretórios e arquivos).

Ao tentar excluir um diretório sem usar `-r`, o sistema exibe um erro:

```
[root@localhost 4linux]# rm novacopia/
rm: cannot remove 'novacopia/': Is a directory
[root@localhost 4linux]# rm -r novacopia/
```

É necessário adicionar **-r**, como no segundo comando, para que a exclusão tenha efeito.

> Você deve estar se perguntando por que não usamos `rmdir` neste caso. Existe uma diferença sutil entre os dois comandos. `rmdir` terá sucesso na exclusão apenas se o diretório fornecido estiver vazio, enquanto `rm -r` pode ser usado independentemente de o diretório estar vazio ou não.

Adicione a opção **-i** para pedir a confirmação antes que o arquivo seja excluído:

`shell [root@localhost 4linux]# rm -ri 4linux rm: descend into directory '4linux'?`# Criando Hard e Soft links ## Pontos de estudo para o exame Os candidatos ao exame `RHCSA` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - **Crie links físicos e virtuais**
  - Liste, defina e altere as permissões padrão ugo/rwx
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Links

No Linux, você pode usar um recurso chamado "links". Os links permitem que os usuários editem o mesmo arquivo em locais diferentes. Existem 2 tipos de links:

- Links simbólicos (também conhecidos como links simbólicos ou Soft Links).
- Links duros (ou Hard Links).

Ambos os tipos são links criados com o uso do comando `ln`. Para entender como os links simbólicos e físicos diferem um do outro, você precisa primeiro entender o que são **inodes**.

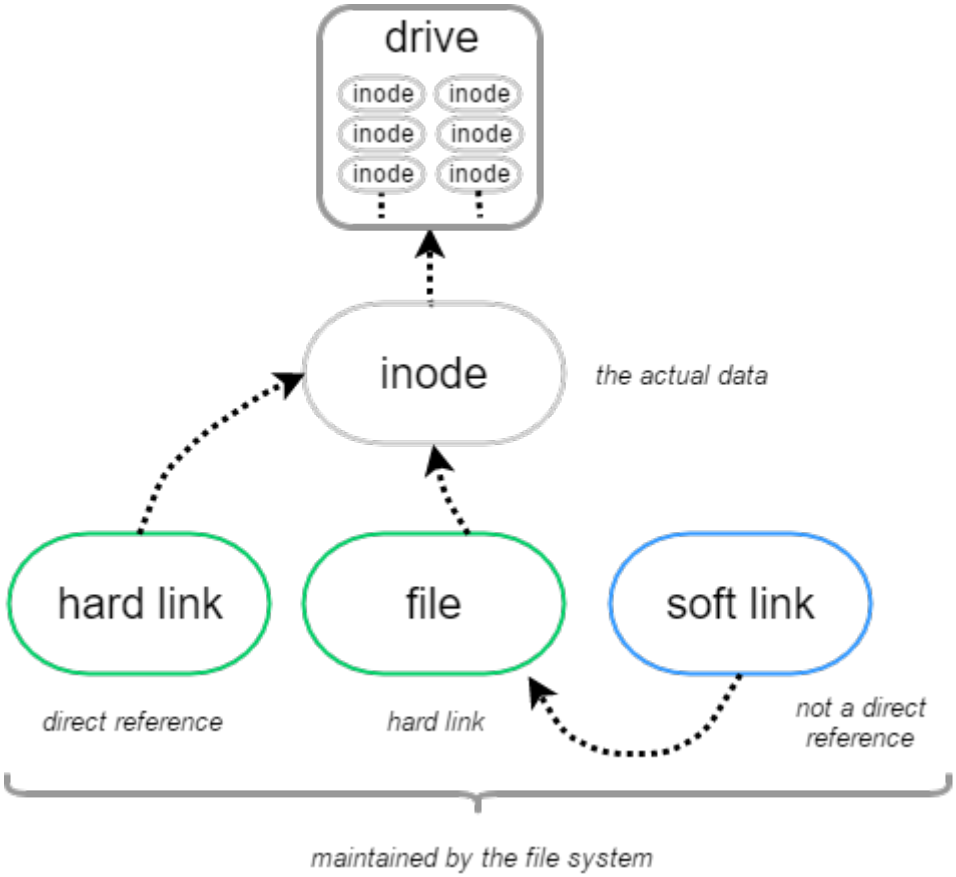


Fig. 4.1: inodes

inodes

Um inode (também conhecido como "nó de índice") é uma entrada em uma tabela de sistema de arquivos que faz referência a um local em um sistema de arquivos. Em outras palavras, é como se um livro de referência tivesse um índice na parte de trás, contendo uma lista de chaves junto com os números das páginas. Assim, um sistema de arquivos também tem um índice, mas, em vez de as chaves serem palavras-chave, são os caminhos absolutos para os arquivos e, em vez de números de página, lista valores de inode.

Você pode pesquisar o valor do inode para um arquivo específico usando a opção `-i` do comando `ls`:

```
[root@localhost 4linux]# echo "Joatham 4Linux" > /tmp/testfile.txt ; ls -li /tmp/testfile.
    txt
```

Neste exemplo, a chave é `/tmp/testfile.txt` e seu valor de inode é `2917`

Observe também a terceira coluna, que mostra o numeral `1`. Isso diz quantas entradas no índice apontam para o mesmo valor de inode.

Hard Links

Essa analogia é como os links físicos funcionam: imagine os links físicos como a entrada do índice, enquanto os valores de inode são os números das páginas. Portanto, criar um link físico é como criar uma nova entrada no índice, mas faz referência ao mesmo valor de inode (também conhecido como número da página). Veja como criar um link físico:

```
[root@localhost 4linux]# ln file /tmp/file ; ls -la /tmp
```

Como você pode observar, o valor anterior `1` agora foi incrementado para `2`. Os links físicos têm as seguintes características:

- Todos os diretórios recém-criados sempre começam com um valor de inode de `2`, isso porque o diretório . que é criado automaticamente dentro da pasta é na verdade um hardlink gerado automaticamente.
- Não é possível criar um link físico para uma pasta . . . você receberá uma mensagem de erro se tentar.
- Você só pode criar links físicos para arquivos para os quais, pelo menos, tenha permissão de leitura, por razões de segurança.

Soft-links

Comparados aos links físicos, os links virtuais funcionam mais como os atalhos do Microsoft Windows. Os links virtuais basicamente redirecionam você para o arquivo de origem. Isso significa que os links virtuais são quebrados se tentar mover o arquivo-fonte para um local diferente ou renomear esse arquivo. Você cria links simbólicos usando o comando `ln` novamente junto com a `s` de **S**oft link habilitada:

```
[root@localhost 4linux]# ln -s file /tmp/file ; ls -la /tmp
```

Observe que os links simbólicos são denotados por `l`, além de prefixados às permissões de outros grupos de usuários. Observe também que o valor do inode é diferente do arquivo original. Observe também que o nome do arquivo possui uma seta apontando para o arquivo

real. Por fim, por padrão, os links simbólicos têm permissões 777. Isso ocorre porque as permissões reais são gerenciadas pelo arquivo real para o qual o link virtual está apontando.

O valor do link do arquivo é incrementado em 1, uma vez que os links flexíveis são coisas conceitualmente diferentes dos links físicos, conforme indicado. Os links flexíveis têm valores de inode diferentes em comparação com o valor de inode do arquivo de origem. Como os soft-links fazem referência à chave de outro arquivo, isso significa que o soft-link será quebrado se você renomear ou mover o arquivo de origem.

Os softlinks têm duas vantagens:

- Os soft links funcionam em sistemas de arquivos (hdd, partições, ... etc);
- Links simbólicos podem ser usados para vincular diretórios. # Listar, definir e alterar permissões padrão ugo/rwx ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:
- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - **Liste, defina e altere as permissões padrão ugo/rwx**
  - Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Introdução

Por ser um sistema multiusuário, o Linux precisa de alguma forma rastrear quem é o proprietário de cada arquivo, e se um usuário tem ou não permissão para executar ações nele. Isso serve para garantir a privacidade dos usuários que desejam manter o conteúdo de seus arquivos em sigilo, bem como para permitir colaboração, tornando certos arquivos acessíveis a diversos usuários.

Esse processo acontece por meio de um sistema de permissões em três níveis. Cada arquivo em disco pertence a um usuário e a um grupo de usuários. Além disso, existem três conjuntos de permissões: um para seu proprietário, um para o grupo que possui o arquivo e um para

todos os outros. Nesta lição, você aprenderá a consultar as permissões de um arquivo, o significado dessas permissões e como manipulá-las.

Consulta de informações sobre arquivos e diretórios

O comando **ls** é usado para obter uma lista do conteúdo de qualquer diretório. No entanto, há muito mais informações disponíveis para cada arquivo, incluindo seu tipo, tamanho, propriedade e muito mais. Para visualizar essas informações, você deve pedir ao `ls` uma lista de "formato longo", usando o parâmetro `-l`. Veja:

```
[root@localhost 4linux]# ls -l
total 536
drwxrwxr-x 2 4linux 4linux   4096 Jun 10 15:57 diretorio_qualquer
-rw------- 1 4linux 4linux 539663 Jun 10 10:43 foto.jpg
-rw-rw-r-- 1 4linux 4linux   1881 Jun 10 15:57 texto.txt
```

Cada coluna da saída acima tem um significado. Vamos dar uma olhada nas colunas relevantes para esta lição:

- A primeira coluna da lista mostra o tipo de arquivo e as permissões. Por exemplo, em drwxrwxr-x:
  - O primeiro caractere, d, indica o tipo de arquivo;
  - Os três caracteres seguintes, rwx, indicam as permissões do proprietário do arquivo, também chamado de usuário ou u;
  - Os três caracteres seguintes, rwx, indicam as permissões do grupo que possui o arquivo, também chamado de g;
  - Os três últimos caracteres, r-x, indicam as permissões de todos os outros ou o.

> Também é comum chamar as permissões de outros de permissões world (mundo), como em "Todo mundo tem essas permissões".

- A terceira e quarta colunas mostram informações sobre a propriedade: respectivamente o usuário e o grupo que possuem o arquivo;
- A sétima e a última colunas mostram o nome do arquivo.

A segunda coluna indica o número de links físicos que apontam para aquele arquivo. A quinta coluna mostra o tamanho do arquivo. A sexta coluna mostra a data e hora em que o arquivo foi modificado pela última vez. Mas essas colunas não são relevantes para o tópico atual.

E quanto aos diretórios?

Se você solicitar informações sobre um diretório usando **ls -l**, ele mostra uma lista do conteúdo do diretório:

```
[root@localhost 4linux]# ls -l diretorio_qualquer/
total 0
-rw-r--r-- 1 4linux 4linux 0 Jun 10 17:59 outro_arquivo.txt
```

Para evitar isso e consultar informações sobre o próprio diretório, adicione o parâmetro -d a ls:

```
[root@localhost 4linux]# ls -l -d diretorio_qualquer/
drwxrwxr-x 2 4linux 4linux 4096 Jun 10 17:59 diretorio_qualquer/
```

Exibindo arquivos ocultos

A listagem do diretório que recuperamos usando **ls -l** anteriormente está incompleta:

```
[root@localhost 4linux]# ls -l
total 544
drwxrwxr-x 2 4linux 4linux   4096 Jun 10 17:59 diretorio_qualquer
-rw------- 1 4linux 4linux 539663 Jun 10 10:43 foto.jpg
-rw-rw-r-- 1 4linux 4linux   1881 Jun 10 15:57 texto.txt
```

Existem três outros arquivos nesse diretório, mas eles estão ocultos. No Linux, os arquivos cujo nome começa com um ponto (.) são ocultados automaticamente. Para vê-los, precisamos adicionar o parâmetro **-a** ao ls:

```
[root@localhost 4linux]# ls -l -a
total 544
drwxrwxr-x 3 4linux 4linux   4096 Jun 10 16:01 .
drwxrwxr-x 4 4linux 4linux   4096 Jun 10 15:56 ..
drwxrwxr-x 2 4linux 4linux   4096 Jun 10 17:59 diretorio_qualquer
-rw------- 1 4linux 4linux 539663 Jun 10 10:43 foto.jpg
-rw-rw-r-- 1 4linux 4linux   1881 Jun 10 15:57 texto.txt
-rw-r--r-- 1 4linux 4linux      0 Jun 10 16:01 .oculto
```

O arquivo .oculto está oculto simplesmente porque o nome começa com . (ponto).

Os diretórios . e .., porém, são especiais. . é um ponteiro para o diretório atual. E .. é um ponteiro para o diretório pai, aquele que contém o atual. No Linux, cada diretório contém pelo menos esses dois diretórios.

> É possível combinar os parâmetros do ls (e muitos outros comandos do Linux). ls -l -a, por exemplo, pode ser escrito como ls -la.

Entendendo os tipos de arquivos

Já mencionamos que a primeira letra em cada saída de ls -l descreve o tipo do arquivo. Os três tipos de arquivo mais comuns são:

- **- (arquivo normal)**: um arquivo pode conter dados de qualquer tipo e ajuda a gerenciar esses dados. Os arquivos podem ser modificados, movidos, copiados e excluídos;
- **d (diretório)**: um diretório contém outros arquivos ou diretórios e ajuda a organizar o sistema de arquivos. Tecnicamente, os diretórios são um tipo especial de arquivo;
- **l (link simbólico)**: este "arquivo" é um ponteiro para outro arquivo ou diretório em outro local no sistema de arquivos.

Além desses, existem três outros tipos de arquivo que você precisa pelo menos conhecer, mas estão fora do escopo desta lição:

- **b (dispositivo de bloco)**: este arquivo representa um dispositivo virtual ou físico, geralmente discos ou outros tipos de dispositivos de armazenamento, como o primeiro disco rígido, que pode ser representado por /dev/sda;
- **c (dispositivo de caracteres)**: este arquivo representa um dispositivo virtual ou físico. Terminais (como o terminal principal em /dev/ttyS0) e portas seriais são exemplos comuns de dispositivos de caracteres;
- **s (socket)**: sockets servem como "canais" passando informações entre dois programas.

> Não altere nenhuma permissão nos dispositivos de bloco, dispositivos de caracteres ou sockets, a menos que saiba muito bem o que está fazendo. Isso pode fazer o sistema parar de funcionar!

Entendendo as permissões

Na saída de ls -l, as permissões de arquivo são mostradas logo após o tipo de arquivo, como três grupos de três caracteres cada, na ordem r, w e x. Eis o que significam. Lembre-se de que um traço - representa a falta de permissão.

Permissões de arquivos

- **r**: significa read (leitura) e tem um valor octal de 4 (não se preocupe, falaremos de octais em breve). Indica permissão para abrir um arquivo e ler seu conteúdo;
- **w**: significa write (escrita) e tem um valor octal de 2. Indica permissão para editar ou excluir um arquivo;
- **x**: significa execute (execução) e tem um valor octal de 1. Indica que o arquivo pode ser executado como um executável ou script.

Assim, por exemplo, um arquivo com permissões rw- pode ser lido e escrito, mas não pode ser executado.

Permissões em diretórios

- **r**: significa read (leitura) e tem um valor octal de 4. Indica permissão para ler o conteúdo do diretório, como nomes de arquivos. Mas não implica em permissão para ler os arquivos em si;
- **w**: significa write (escrita) e tem um valor octal de 2. Indica permissão para editar ou excluir arquivos em um diretório, ou alterar seus nomes, permissões e proprietários. Se um usuário tiver a permissão w em um diretório, ele poderá alterar as permissões de qualquer arquivo dentro do diretório (o conteúdo do diretório), mesmo que o usuário não tenha permissões no arquivo ou se o arquivo pertencer a outro utilizador.

Lembre-se de que ter permissões de gravação em um diretório ou arquivo não significa que você tem permissão para remover ou renomear o diretório ou arquivo em si.

- **x**: significa execute (execução) e tem um valor octal de 1. Indica permissão para entrar em um diretório, mas não para listar seus arquivos (para isso, r é necessário).

A última parte sobre diretórios pode parecer um pouco confusa. Vamos imaginar, por exemplo, que você tem um diretório chamado diretorio_qualquer, com as seguintes permissões:

```
[root@localhost 4linux]# ls -ld diretorio_qualquer/
d--x--x--x 2 4linux 4linux 4,0K Jun 20 18:46 diretorio_qualquer
```

Imagine também que dentro deste diretório há um script de shell chamado oi.sh:

```
-rwxr-xr-x 1 4linux 4linux 33 Jun 20 18:46 oi.sh
```

Se você for a usuário 4linux e tentar listar o conteúdo de diretorio_qualquer, receberá uma mensagem de erro, pois seu usuário não tem permissão de leitura para esse diretório:

```
[root@localhost 4linux]# ls -l diretorio_qualquer/
ls: cannot open directory 'diretorio_qualquer/': Permission denied
```

No entanto, o usuário 4linux tem permissões de execução, o que significa que ele pode entrar no diretório. Portanto, o usuário 4linux pode acessar arquivos dentro do diretório, desde que tenha as permissões corretas para o respectivo arquivo. Vamos supor que o usuário tem permissões totais (rwx) para o script oi.sh. Nesse caso, se souber o nome do arquivo completo, ele pode executar o script, embora não possa ler o conteúdo do diretório que o contém:

```
[root@localhost 4linux]# sh diretorio_qualquer/oi.sh
Salve Galeraa!!!
```

Como dissemos antes, as permissões são especificadas em sequência: primeiro para o proprietário do arquivo, depois para o grupo proprietário e, em seguida, para outros usuários. Sempre que alguém tenta realizar uma ação no arquivo, as permissões são verificadas na mesma ordem.

Primeiro, o sistema verifica se o usuário atual possui o arquivo e, se for o caso, ele aplica apenas o primeiro conjunto de permissões. Caso contrário, ele verifica se o usuário atual pertence ao grupo que possui o arquivo. Nesse caso, ele aplica o segundo conjunto de permissões apenas. Em qualquer outro caso, o sistema aplicará o terceiro conjunto de permissões.

Isso significa que, se o usuário atual for o proprietário do arquivo, apenas as permissões do proprietário serão efetivas, mesmo se as permissões do grupo ou outras forem mais permissivas do que as do proprietário.

Modificando as permissões de arquivos

O comando chmod é usado para modificar as permissões de um arquivo. Ele pede pelo menos dois parâmetros: o primeiro descreve quais permissões alterar, o segundo aponta para o arquivo ou diretório onde a alteração será feita. Lembre-se de que apenas o proprietário do arquivo ou o administrador do sistema (root) pode alterar as permissões em um arquivo.

As permissões a alterar podem ser descritas de duas maneiras, ou "modos", diferentes.

O primeiro, denominado modo simbólico, oferece um controle refinado, permitindo adicionar ou revogar uma única permissão sem modificar as outras no conjunto. O outro modo, chamado

modo octal, é mais fácil de lembrar e mais rápido de usar quando desejamos definir todos os valores de permissão de uma vez.

Ambos os modos levam ao mesmo resultado final. Assim, por exemplo, os comandos:

```
[root@localhost 4linux]# chmod ug+rw-x,o-rwx texto.txt
```

e

```
[root@localhost 4linux]# chmod 660 texto.txt
```

produzem exatamente a mesma saída, um arquivo com as permissões definidas:

```
-rw-rw---- 1 4linux 4linux  765 Jun 20 21:25 texto.txt
```

Agora, vamos ver como cada modo funciona.

Modo simbólico

Ao descrever quais permissões alterar no modo simbólico, o(s) primeiro(s) caractere(s) indica(m) as permissões que serão alteradas: de usuário (u), grupo (g), outros (o) e/ou todos (a).

Então você precisa dizer ao comando o que fazer: você pode conceder uma permissão (+), revogar uma permissão (-) ou defini-la com um valor específico (=).

Por último, você especifica em qual permissão deseja agir: leitura (r), escrita (w) ou execução (x).

Por exemplo, imagine que temos um arquivo chamado texto.txt com o seguinte conjunto de permissões:

```
[root@localhost 4linux]# ls -l texto.txt
-rw-r--r-- 1 4linux 4linux 765 Jun 20 21:25 texto.txt
```

Se você deseja conceder permissões de gravação aos membros do grupo proprietário do arquivo,

deve usar o parâmetro g+w. É mais fácil pensar desta forma: "Para o grupo (g), conceda (+) permissões de escrita (w)". Então, o comando seria:

```
[root@localhost 4linux]# chmod g+w texto.txt
```

Vamos conferir o resultado com ls:

```
[root@localhost 4linux]# ls -l texto.txt
-rw-rw-r-- 1 4linux 4linux 765 Jun 20 21:25 texto.txt
```

Deseja remover as permissões de leitura para o proprietário do mesmo arquivo? Pense assim: "Para o usuário (u), revogue (-) as permissões de leitura (r)". Portanto, o parâmetro é u-r, desta maneira:

```
[root@localhost 4linux]# chmod u-r texto.txt
[root@localhost 4linux]# ls -l texto.txt
--w-rw-r-- 1 4linux 4linux 765 Jun 20 21:25 texto.txt
```

E se quisermos definir as permissões exatamente como rw- para todos? Nesse caso, pensamos assim: "Para todos (a), defina exatamente (=) leitura (r), escrita (w), e não execução (-)". Assim:

```
[root@localhost 4linux]# chmod a=rw- texto.txt
[root@localhost 4linux]# ls -l texto.txt
-rw-rw-rw- 1 4linux 4linux 765 Jun 20 21:25 texto.txt
```

Claro, é possível modificar diversas permissões ao mesmo tempo. Neste caso, separe-os com uma vírgula (,):

```
[root@localhost 4linux]# chmod u+rwx,g-x texto.txt
[root@localhost 4linux]# ls -lh texto.txt
-rwxrw-rw- 1 4linux 4linux 765 Jun 20 21:25 texto.txt
```

O exemplo acima pode ser lido como: "Para o usuário (u), conceda (+) permissões de leitura, escrita e execução (rwx), para o grupo (g), revogue (-) permissões de execução (x)".

Quando executado em um diretório, chmod modifica apenas as permissões do diretório. chmod também possui um modo recursivo, útil quando desejamos alterar as permissões para "todos os arquivos dentro de um diretório e seus subdiretórios". Para usá-lo, adicione o parâmetro -R após o nome do comando, antes das permissões a alterar:

```
[root@localhost 4linux]# chmod -R u+rwx diretorio_qualquer/
```

Este comando pode ser lido como: "Recursivamente (-R), para o usuário (u), conceda (+) permissões de leitura, escrita e execução (rwx)".

> Tenha cuidado e pense duas vezes antes de usar a opção -R, pois é fácil alterar sem querer as permissões de arquivos e diretórios, especialmente em diretórios com um grande número de arquivos e subdiretórios.

Modo octal

No modo octal, as permissões são especificadas de maneira diferente: como um valor de três dígitos na notação octal, um sistema numérico de base 8.

Cada permissão tem um valor correspondente e elas são especificadas na seguinte ordem: primeiro vem leitura (r), que é 4, depois escrita (w), que é 2, e por fim execução (x), representada por 1. Se não houver permissão, usamos o valor zero (0). Portanto, uma permissão rwx seria 7 (4 + 2 + 1) e rx seria 5 (4 + 0 + 1).

O primeiro dos três dígitos no conjunto de permissões representa as permissões do usuário (u), o segundo as do grupo (g) e o terceiro as do outros (o). Se quisermos definir as permissões de um arquivo como rw-rw—-, o valor octal seria 660:

```
[root@localhost 4linux]# chmod 660 texto.txt
[root@localhost 4linux]# ls -l texto.txt
-rw-rw---- 1 4linux 4linux 765 Jun 20 21:25 texto.txt
```

Além disso, a sintaxe no modo octal é a mesma que no modo simbólico: o primeiro parâmetro representa as permissões que você deseja alterar e o segundo aponta para o arquivo ou diretório onde a alteração será feita.

> Se o valor de uma permissão for ímpar, o arquivo certamente é executável!

Qual sintaxe você deve usar? O modo octal é recomendado quando se deseja alterar as permissões para um valor específico, por exemplo 640 (rw- r– —).

O modo simbólico é mais útil se você deseja inverter apenas um valor específico, independentemente das permissões atuais do arquivo. Por exemplo, você pode adicionar permissões de execução para o usuário usando apenas chmod u+x script.sh sem levar em conta, ou mesmo tocar, as permissões atuais de grupo e outros.

Modificando o proprietário de um arquivo

O comando `chown` é usado para modificar a propriedade de um arquivo ou diretório. A sintaxe é bastante simples:

```
chown USERNAME:GROUPNAME FILENAME
```

Por exemplo, vamos verificar um arquivo chamado texto.txt:

```
[root@localhost 4linux]# ls -l texto.txt
-rw-rw---- 1 4linux 4linux 1881 Jun 10 15:57 texto.txt
```

O usuário que possui o arquivo é 4linux, e o grupo também é 4linux. Agora, vamos mudar o grupo proprietário do arquivo para um outro grupo, como students:

```
[root@localhost 4linux]# chown 4linux:students texto.txt
[root@localhost 4linux]# ls -l texto.txt
-rw-rw---- 1 4linux students 1881 Jun 10 15:57 texto.txt
```

Permissões padrão

Vamos fazer uma experiência... Abra uma janela de terminal e crie um arquivo vazio com o seguinte comando:

```
[root@localhost 4linux]# touch testfile
```

Agora, vamos dar uma olhada nas permissões deste arquivo. Eles podem ser diferentes em seu sistema, mas vamos supor que sejam mais ou menos assim:

```
[root@localhost 4linux]# ls -lh testfile
-rw-r--r-- 1 4linux 4linux 0 jul 13 21:55 testfile
```

As permissões são **rw-r—r–**: leitura e escrita para o usuário, e leitura para o grupo e outros, ou 644 no modo octal. Agora, tente criar um diretório:

```
[root@localhost 4linux]# mkdir testdir
[root@localhost 4linux]# ls -lhd testdir
drwxr-xr-x 2 4linux 4linux 4,0K jul 13 22:01 testdir
```

Agora as permissões são **rwxr-xr-x**: leitura, escrita e execução para o usuário, leitura e execução para o grupo e outros, ou 755 no modo simbólico.

Não importa onde você esteja no sistema de arquivos, cada arquivo ou diretório que criar terá as mesmas permissões. Você já se perguntou de onde elas vêm?

A resposta é: da máscara de usuário ou **umask**, que define as permissões padrão para cada arquivo criado. Você pode verificar os valores atuais com o comando umask:

```
[root@localhost 4linux]# umask
0022
```

Mas isso não se parece com **rw-r—r–**, ou mesmo 644. Talvez devêssemos tentar com o parâmetro -S, para obter uma saída em modo simbólico:

```
[root@localhost 4linux]# umask -S
u=rwx,g=rx,o=rx
```

Essas são as mesmas permissões que nosso diretório de teste obteve em um dos exemplos acima. Mas por que, quando criamos um arquivo, as permissões eram diferentes?

Bem, não faz sentido definir permissões globais de execução para todos em qualquer arquivo por padrão, certo? Os diretórios precisam de permissões de execução (caso contrário, não é possível entrar neles), mas os arquivos não, então eles não as recebem. Daí o **rw-r—r–**.

Além de exibir as permissões padrão, o umask também pode ser usado para alterá-las para sua sessão do shell atual. Por exemplo, se usarmos o comando:

```
[root@localhost 4linux]# umask u=rwx,g=rwx,o=
```

Cada novo diretório herdará as permissões **rwxrwx—**, e cada arquivo **rw-rw—**- (já que eles não recebem permissões de execução). Se você repetir os exemplos acima para criar um testfile e testdir e verificar as permissões, o resultado será:

```
[root@localhost 4linux]# ls -lhd test*
drwxrwx--- 2 4linux 4linux 4,0K jul 13 22:25 testdir
-rw-rw---- 1 4linux 4linux    0 jul 13 22:25 testfile
```

E se você marcar o **umask** sem o parâmetro -S (modo simbólico), você obterá:

```
[root@localhost 4linux]# umask
0007
```

O resultado não parece familiar porque os valores usados são diferentes. Eis uma tabela com todos os valores e seus respectivos significados:

Valores	Permissão dos Arquivos	Permissões dos diretórios
0	rm-	rwx
1	rw	rw-
2	r–	r-x
3	r–	r–
4	-w-	-wx
5	-w-	-w-
6	—	–x
7	—	—

Como vemos, 007 corresponde a **rwxrwx—**, exatamente como solicitamos. O zero inicial pode ser ignorado.

Permissões especiais

Além das permissões de leitura, gravação e execução para usuário, grupo e outros, cada arquivo pode ter três outras permissões especiais capazes de alterar a maneira como um diretório funciona ou como um programa é executado. Elas podem ser especificadas no modo simbólico ou octal e são as seguintes:

Sticky Bit

O sticky bit, também chamado de sinalizador de exclusão restrito, tem o valor octal 1 e no modo simbólico é representado por um t dentro das permissões de outros. Ele se aplica apenas a diretórios e não tem efeito em arquivos normais. No Linux, ele evita que os usuários removam ou renomeiem um arquivo em um diretório, a menos que sejam proprietários desse arquivo ou diretório.

Os diretórios com o sticky bit definido mostram um `t` substituindo o `x` nas permissões de outros na saída de `ls -l`:

```
[root@localhost 4linux]# ls -ld Diretorio_exemplo/
drwxr-xr-t 2 4linux 4linux 4096 Jun 20 18:46 Diretorio_exemplo/
```

No modo octal, as permissões especiais são especificadas usando uma notação de 4 dígitos, sendo que o primeiro dígito representa a permissão especial sobre a qual agir. Por exemplo, para definir o sticky bit (valor 1) para o diretório diretorio_qualquer no modo octal, com permissões 755, o comando seria:

```
[root@localhost 4linux]# chmod 1755 diretorio_qualquer
[root@localhost 4linux]# ls -ld diretorio_qualquer
drwxr-xr-t 2 4linux 4linux 4,0K Jun 20 18:46 diretorio_qualquer
```

Set GID

O Set GID, também conhecido como `SGID` ou Set Group ID bit, tem o valor octal 2 e no modo simbólico é representado por um s nas permissões de grupo. Ele pode ser aplicado a arquivos executáveis ou diretórios. Nos arquivos, fará com que o processo seja executado com os privilégios do grupo que possui o arquivo. Quando aplicado a diretórios, fará com que cada arquivo ou diretório criado herde o grupo do diretório pai.

Arquivos e diretórios com o bit `SGID` mostram um s no lugar do x nas permissões de grupo na

saída de ls -l:

```
[root@localhost 4linux]# ls -l test.sh
-rwxr-sr-x 1 4linux 4linux 33 Jun 11 10:36 test.sh
```

Para adicionar permissões SGID a um arquivo no modo simbólico, o comando seria:

```
[root@localhost 4linux]# chmod g+s test.sh
[root@localhost 4linux]# ls -l test.sh
-rwxr-sr-x 1 4linux root     33 Jun 11 10:36 test.sh
```

O exemplo a seguir o ajudará a entender melhor os efeitos do SGID em um diretório. Suponha que temos um diretório chamado Diretorio_exemplo, pertencente à usuário 4linux e ao grupo users, com a seguinte estrutura de permissões:

```
[root@localhost 4linux]# ls -ldh Diretorio_exemplo/
drwxr-xr-x 2 4linux users 4,0K Jan 18 17:06 Diretorio_exemplo/
```

Agora, vamos mudar para esse diretório e, usando o comando touch, criar um arquivo vazio dentro dele. O resultado seria:

```
[root@localhost 4linux]# cd Diretorio_exemplo/
[root@localhost 4linux]# touch novoarquivo
[root@localhost 4linux]# ls -lh novoarquivo
-rw-r--r-- 1 4linux 4linux 0 Jan 18 17:11 novoarquivo
```

Como podemos ver, o arquivo é propriedade do usuário 4linux e do grupo 4linux. Mas, se o diretório tivesse a permissão SGID definida, o resultado seria diferente. Primeiro, vamos adicionar o bit SGID ao Diretorio_exemplo e verificar os resultados:

```
[root@localhost 4linux]# sudo chmod g+s Diretorio_exemplo/
[root@localhost 4linux]# ls -ldh Diretorio_exemplo/
drwxr-sr-x 2 4linux users 4,0K Jan 18 17:17 Diretorio_exemplo/
```

O s nas permissões do grupo indica que o bit SGID está definido. Agora, vamos mudar para este diretório e, novamente, criar um arquivo vazio com o comando touch:

```
[root@localhost 4linux]# cd Diretorio_exemplo/
[root@localhost 4linux]# touch arquivovazio
[root@localhost 4linux]# ls -lh arquivovazio
 -rw-r--r-- 1 4linux users 0 Jan 18 17:20 arquivovazio
```

O grupo que possui o arquivo é users. Isso ocorre porque o bit `SGID` fez o arquivo herdar o proprietário do grupo de seu diretório pai, que é users.

Set UID

SUID, também conhecido como Set User ID, tem valor octal 4 e é representado por um s nas permissões de usuário no modo simbólico. Aplica-se apenas aos arquivos e não tem efeito em diretórios. Seu comportamento é semelhante ao do bit `SGID`, mas o processo será executado com os privilégios do usuário proprietário do arquivo. Os arquivos com o bit SUID mostram um `s` no lugar do `x` nas permissões do usuário, na saída de `ls -l`:

```
[root@localhost 4linux]# ls -ld test.sh
-rwsr-xr-x 1 4linux 4linux 33 Jun 11 10:36 test.sh
```

Podemos combinar diversas permissões especiais em um parâmetro somando-as. Assim, para definir o `SGID` (valor 2) e o SUID (valor 4) no modo octal para o script test.sh com permissões 755, digite:

```
[root@localhost 4linux]# chmod 6755 test.sh
```

E o resultado seria:

```
[root@localhost 4linux]# ls -lh test.sh
-rwsr-sr-x 1 4linux 4linux 66 Jan 18 17:29 test.sh
```

5

Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Compreenda e use ferramentas essenciais**
  - Acesse um prompt de shell e emita comandos com a sintaxe correta
  - Use o redirecionamento de entrada-saída (>, », |, 2>, etc.)
  - Use grep e expressões regulares para analisar o texto
  - Acesse sistemas remotos usando SSH
  - Faça login e troque de usuários em destinos multiusuário
  - Arquive, compacte e descompacte arquivos usando tar, star, gzip e bzip2
  - Crie e edite arquivos de texto
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie links físicos e virtuais
  - Liste, defina e altere as permissões padrão ugo/rwx

- **Localize, leia e use a documentação do sistema, incluindo man, informações e arquivos em /usr/share/doc**

Introdução

O ritmo de geração de conhecimento e informação tem sido vertiginoso nos últimos **sessenta anos**, especialmente na área tecnológica. Por isso, é fundamental saber onde buscar informações para manter-se sempre atualizado. Neste capítulo, vamos aprender a consultar as documentações existentes e como buscar informações sobre o que precisamos.

O Sistema Operacional **GNU/Linux** possui uma vasta biblioteca de documentação. Antes de recorrermos a ajuda de outras pessoas, devemos lembrar que podemos ter a respostas que precisamos no próprio sistema, bem a nossa frente, ao teclar de um simples comando. Essa documentação em grande parte dos casos é de extrema qualidade.

O **GNU/Linux** cresceu porque a comunidade contribui para o sistema e sua documentação. Essa comunidade não tem medo ou receio de compartilhar informações e disponibilizar o que foi desenvolvido no próprio sistema. É muito importante reforçar que, no Software Livre, as pessoas nunca ocultam seu **know-how**, ou seja, você pode perguntar à vontade, desde que saiba o que e onde perguntar.

A documentação do GNU/Linux pode ser vista também como fonte de conhecimento, onde pode-se aprender muito sobre cada um dos serviços e comandos disponíveis.

Essa ajuda é provida por meio dos manuais, as famosas **Man Pages**.

Abaixo, vamos começar a nos familiarizar com a documentação existente e as formas nas quais ela é apresentada.

Formas de documentação

Existem diversas formas de se documentar um projeto, dentre elas temos os **How-to's**, os **manuais** e as **documentações**.

How-to's

Os **How-to's** (do inglês "Como Fazer") são documentos que enfocam uma necessidade específica, como montar um "firewall", instalar uma "webcam", configurar placas de som, configurar um servidor web e muitos outros. Normalmente esses documentos são instalados junto com suas respectivas aplicações ou podem ter um pacote específico para a documentação daquela aplicação. Os **how-to's** também são conhecidos como **cook-books** - livro de receitas.

O diretório de **How-to's** do GNU/Linux é o `/usr/share/doc`. Se desejamos saber como configurar um **firewall**, podemos consultar os arquivos do diretório:

```
[root@localhost 4linux]# cd /usr/share/doc/iptables/
```

Muitas vezes o uso de **how-to's** ou **cook-book's** não agrega um bom conhecimento, pois trata-se somente de uma lista de afazeres para chegar a um objetivo. Quando o software é atualizado, todo aquele conhecimento fica dependente de um novo **how-to**.

Manuais

Diferente dos **How-to's**, os manuais não vão te mostrar um passo a passo ou mesmo te dar uma lista de afazeres. O principal objetivo do manual é mostrar como as funcionalidades daquele software podem ser usadas. Com o manual, o aprendizado para a utilização da ferramenta é facilitado, já que possui alguns exemplos de usabilidade. Esses manuais podem ser encontrados através do comando **man**, o qual veremos ainda nesse capítulo, um pouco mais adiante.

Documentação

A palavra documentação é muito intensa. Quando falamos em documentar uma ferramenta, estamos na realidade abrangendo uma série de outros itens importantes, dentre eles os **How-to's** e os manuais. Com a documentação de um projeto é possível entender absolutamente tudo sobre ele, ou seja, essa documentação deve mostrar todas as partes relacionadas ao projeto.

Podemos, por exemplo, citar a documentação de um projeto de rede, onde deve constar não só documentos como **how-to's** e manuais, mas sim todas as especificações dos componentes, bem como cabos, "switch's" e "routers" dentre outros detalhes muito importantes.

Como esse tipo de documentação é muito especifica, devemos consultar o site de cada projeto individualmente.

Existem diversos comandos de ajuda no GNU/Linux, vamos abordar cada um deles logo abaixo:

Comando `help`

O comando `help` provê ajuda para comandos internos do interpretador de comandos, ou seja, o comando `help` fornece ajuda rápida. Ele é muito útil para saber quais opções podem ser

usadas com os comandos internos do interpretador de comandos (shell). Para visualizar uma ajuda rápida para todos os comandos internos do sistema, podemos fazer da seguinte forma:

```
[root@localhost 4linux]# help
```

Caso desejemos visualizar a ajuda rápida para somente um comando interno, usamos esta outra sintaxe:

```
[root@localhost 4linux]# help [comando]
```

> O comando `help` somente mostra a ajuda para comandos internos.

```
[root@localhost 4linux]# help type
```

O comando `type` mostra se cada nome de comando é um comando do UNIX, um comando interno, um alias, uma palavra-chave do shell ou uma função de shell definida.

Verifique o tipo do comando help que conheceremos a seguir:

```
[root@localhost 4linux]# help help
```

Para comandos externos, o `help` aparece como parâmetro. Por exemplo:

```
[comando] --help
```

Desse modo, caso desejemos visualizar uma ajuda rápida sobre um comando externo, faremos da seguinte forma:

```
[root@localhost 4linux]# ls --help
```

O parâmetro `--help` pode ser utilizado em qualquer comando para ter uma consulta rápida

dos parâmetros que ele pode nos oferecer. É importante entender que `--help` é na verdade um parâmetro individual de cada comando. Logo, se um comando não tiver esse parâmetro, existem outros meios para se obter ajuda. Não se esqueça de estudar as diferenças entre comandos internos e externos.

Comando apropos

O comando `apropos` ajuda o usuário quando ele não se lembra do comando exato, mas conhece algumas palavras-chave relacionadas ao comando que definem seu uso ou funcionalidade. Ele pesquisa a página de manual do Linux com a ajuda da palavra-chave fornecida pelo usuário para encontrar o comando e suas funções.

Sintaxe:

```
apropos [OPÇÃO ..] PALAVRA-CHAVE ...
```

Situacao 1: suponhamos que você não saiba como compactar um arquivo, então você poderia digitar o seguinte comando no terminal e ele mostrará todos os comandos relacionados e sua breve descrição ou funcionalidade.

```
[root@localhost 4linux]# apropos compress
```

Depois de executar o comando acima, você observará uma série de comandos listados no terminal que tratam não apenas de como compactar um arquivo, mas também de expandir um arquivo compactado, pesquisar um arquivo compactado, comparar um arquivo compactado etc.

Situacao 2: o comando `apropos` também suporta várias palavras-chave se fornecidas como um argumento, ou seja, também podemos fornecer mais de uma palavra-chave para uma busca melhor. Assim, se duas palavras-chave forem fornecidas, o comando `apropos` exibirá toda a lista do comando que contém a primeira palavra-chave em sua descrição de página de manual ou a segunda palavra-chave.

```
[root@localhost 4linux]# apropos email
```

> Entrada 1 (com uma palavra-chave)

```
[root@localhost 4linux]# apropos email address
```

> Entrada 2 (com várias palavras-chave)

E por fim, uma forma equivalente ao `apropos` é usar o comando `man` juntamente com a opção `-k`:

```
[root@localhost 4linux]# man -k editor
```

Comando `whatis`

O comando `whatis` tem basicamente a mesma função do comando `apropos`, porém, as buscas do comando `whatis` são mais específicas. O `apropos` busca as páginas de manuais e descrições de maneira mais genérica. Se digitarmos a palavra **passwd**, ele nos trará tudo que tiver **passwd**, seja como nome ou parte do nome do manual ou na descrição. Já o `whatis` nos trará somente o manual com nome exato da palavra pesquisada.

A sintaxe utilizada no comando `whatis` é a seguinte:

```
whatis [comando]
```

Você sabe que tem um programa chamado `vim`, mas não sabe o que ele faz?

```
[root@localhost 4linux]# whatis vim
```

> Uma forma equivalente ao `whatis` é usar o comando `man` juntamente com a opção `-f`:

```
[root@localhost 4linux]# man -f vim
```

Para localizar as **man pages**, o comando `apropos` e `whatis` utilizam o mesmo banco de dados construído com o comando `catman` ou `makewhatis` (executado pelo administrador do sistema, **root**).

Para construir o banco de dados do comando `apropos` e `whatis`, devemos executar o comando abaixo:

RHEL8:

```
[root@localhost 4linux]# catman
```

Comando man

O comando `man`, sem dúvidas, é o mais usado para obtenção de documentação no Linux. Ele é o responsável por trazer os manuais mais completos sobre determinado comando, arquivo de configuração, bibliotecas, entre outros nos quais estamos trabalhando.

Os manuais do sistema são divididos nos seguintes níveis:

- **man 1** -> Programas e executáveis disponíveis ao usuário;
- **man 2** -> Rotinas de sistema Unix e C;
- **man 3** -> Rotinas de bibliotecas da linguagem C;
- **man 4** -> Arquivos especiais (dispositivos em `/dev`);
- **man 5** -> Arquivos de configuração e convenções;
- **man 6** -> Games;
- **man 7** -> Diversos (macros textuais, por exemplo, `regex`);
- **man 8** -> Comandos administrativos;
- **man 9** -> Rotinas internas do kernel.

> É comum o exame da RHEL cobrar mais dos níveis 1, 5 e 8 dos manuais! Então, lembre-se de estudar binários, arquivos de configuração e comandos administrativos.

Sintaxe do comando `man`:

```
[root@localhost 4linux]# man [ comando ]
```

ou

```
man [ seção ][ comando ]
```

> Uma curiosidade: as informações sobre as seções do comando `man` podem ser encontradas em seu próprio manual, digitando o comando `man man`.

Se for necessário visualizar o manual do comando `passwd`, podemos fazer da seguinte forma:

```
[root@localhost 4linux]# man passwd
```

Para navegar pelo manual, o comando `man` abre um arquivo que está compactado na pasta `/usr/share/man/man1` para o `passwd`. Outros níveis de manuais dependem do comando ou arquivo. O `passwd` é conhecido no sistema **GNU/Linux** como um comando que adiciona ou modifica a senha do usuário e, também, como o arquivo de usuários do sistema `/etc/passwd)`.

Veremos agora o manual do arquivo de usuários `passwd`:

```
[root@localhost 4linux]# man 5 passwd
```

Podemos consultar quais manuais estão disponíveis dentro do próprio diretório do `man`:

```
[root@localhost 4linux]# ls /usr/share/man/
```

Dentro desse diretório é possível ver todas as divisões dos manuais: os níveis, os idiomas e mais. Todos os níveis de manuais possuem sua determinada introdução que pode ser vista com o comando:

```
[root@localhost 4linux]# man <nivel> intro
```

Podemos ver que para visualizar o manual do arquivo de usuário `passwd`, precisamos informar em qual nível de manual ele se encontra, pois já existe um `passwd` no nível 1, que é o comando,

então ele aparece primeiro quando digitamos `man passwd` sem indicar o nível.

Esse manual do arquivo `passwd` está compactado na pasta **/usr/share/man/man5**.

Comando `info`

As **info pages** são como as páginas de manuais, mas são utilizadas com navegação entre as páginas. Elas são acessadas pelo comando `info` - útil quando já sabemos o nome do comando e só queremos saber qual sua respectiva função.

A navegação nas **info pages** é feita através de nomes marcados com um **(*) **(hipertextos) que, ao pressionarmos** Enter**, nos leva até a seção correspondente, e** Backspace** volta à página anterior. Algo parecido com a navegação na Internet.

Podemos também navegar pelas páginas com as teclas: * **n (next/próximo);** * **p (previous/anterior);** * **u (up/sobe um nível)**.

Para sair do comando `info`, basta pressionar a tecla `q`.

Se for necessário exibir a lista de todos os manuais de comandos/programas disponíveis, execute o comando abaixo sem nenhum argumento. Assim:

```
[root@localhost 4linux]# info
```

Para exibir as informações somente de um determinado comando, usaremos a seguinte sintaxe:

```
info [comando]
```

Visualizar informações do comando vim:

```
[root@localhost 4linux]# info vim
```

Comando `whereis`

O comando `whereis` é utilizado para mostrar a localização do binário do comando, do arquivo de configuração (caso exista), e a localização das páginas de manuais do determinado comando ou arquivo.

Se compararmos o comando `whereis` com o comando `find`, eles parecerão semelhantes entre si, pois ambos podem ser usados para os mesmos fins, mas o comando `whereis` produz o resultado com mais precisão, consumindo menos tempo comparativamente.

Exemplo 1: digamos que queremos encontrar a localização do comando `apropos` e, em seguida, precisamos executar o seguinte comando no terminal:

```
[root@localhost 4linux]# whereis apropos
```

Exemplo 2: para encontrar a localização do comando `lshw`.

```
[root@localhost 4linux]# whereis lshw
```

Exemplo 3: Para exibir os arquivos do diretório atual que não possuem arquivo de documentação.

```
[root@localhost 4linux]# whereis -m -u *
```

Exemplo 4: para localizar o binário de lesspipe no caminho, `/bin`.

```
[root@localhost 4linux]# whereis -B /bin -f lesspipe
```

> -**B**: esta opção é usada para alterar ou limitar os locais onde o `whereis` procura por binários.

Exemplo 5: para verificar a página de manual de introdução que está apenas em um local específico, ou seja, `/usr/share/man/man1`.

```
[root@localhost 4linux]# whereis -M /usr/share/man/man1 -f intro
```

> -**M**: esta opção é usada para alterar ou limitar os locais onde o `whereis` procura por seções manuais.

Exemplo 6: para localizar todos os arquivos em `/usr/bin` que não estão documentados em `/usr/man/man1` com fonte em `/usr/src`

```
[root@localhost 4linux]# whereis -u -M /usr/share/man/man1 -S /usr/src -f *
```

> -**S**: esta opção é usada para alterar ou de outra forma limitar os locais onde `whereis` procura pelas fontes.

> -**f**: esta opção simplesmente termina a última lista de diretórios e sinaliza o início dos nomes dos arquivos. Deve ser usado quando qualquer uma das opções `-B` , `-M` ou `-S` for usada. -**V**: exibe informações sobre a versão e sai. -**h** : exibe esta ajuda e sai.

Comando which

O comando `which` é bem semelhante ao comando `whereis`, entretanto, só mostra a localização do binário do comando.

Para visualizar a localização do binário do comando, utilizamos a seguinte sintaxe:

```
[root@localhost 4linux]# which <comando>
```

Localização do binário do comando `vim`:

```
[root@localhost 4linux]# which vim
```

6

Executar código de maneira condicional (com if, test, [] etc.)

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Crie scripts de shell simples**
  - **Executar código condicionalmente (uso de: if, test, [], etc.)**
  - Usar construções de loop (for, etc.) para processar arquivo, entrada de linha de comando
  - Entradas de script de processo (`$1`, `$2`, etc.)
  - Processar saída de comandos shell em um script
  - Processar códigos de saída de comando shell

Execução condicional

Execução condicional significa que você pode optar por executar o código apenas se certas condições forem atendidas. Sem esse recurso, tudo o que você seria capaz de fazer é executar um comando após o outro, após o outro. A capacidade de testar uma variedade de coisas sobre o estado do sistema e das variáveis de ambiente do processo significa que um script de shell pode fazer coisas muito mais poderosas que seriam possíveis de outra forma.

Sintaxe:

```
[ condition-to-test-for ]
```

Syntaxe do comando `test`

```
[root@localhost 4linux]# test condição && true || false
```

Teste se o arquivo existe:

```
[root@localhost 4linux]# test -e /tmp/a && echo exist || echo not exist
```

Teste se numero é maior ou menor que 3:

```
[root@localhost 4linux]# test 7 -gt 3 && echo '> que 3' || echo '< que 3'
[root@localhost 4linux]# test 1 -gt 3 && echo '> que 3' || echo '< que 3'
```

Exemplo:

```
[ -e /etc/passwd ]
```

Isso testa se **etc/passwd** existe e, se existir, retorna true - status de saída do comando de `0`. Se não existir, o comando sai com o status de saída de `1`. Os espaços ao redor dos símbolos `[` e `]`são obrigatórios!

Operadores de teste de arquivo:

- **-d ARQUIVO** - Verdadeiro se o arquivo for um diretório.
- **-e ARQUIVO** - Verdadeiro se o arquivo existir.
- **-f ARQUIVO** - Verdadeiro se o arquivo existir e for um arquivo normal.
- **-r ARQUIVO** - Verdadeiro se o arquivo puder ser lido por você.
- **-s ARQUIVO** - Verdadeiro se o arquivo existir e não estiver vazio.
- **-w ARQUIVO** - Verdadeiro se o arquivo puder ser escrito por você.
- **-x ARQUIVO** - Verdadeiro se o arquivo for executável por você.

Operadores de teste de string:

- **-z STRING** - Verdadeiro se a string estiver vazia.
- **-n STRING** - Verdadeiro se a string não estiver vazia.
- **STRING1 = STRING2** - Verdadeiro se as strings forem iguais.
- **STRING1 != STRING2** - Verdadeiro se as strings não forem iguais.

Testes aritméticos:

- **arg1 -eq arg2** - Verdadeiro se os argumentos forem iguais.
- **arg1 -ne arg2** - Verdadeiro se os argumentos não forem iguais.
- **arg1 -lt arg2** - Verdadeiro se o arg1 for menor que arg2.
- **arg1 -le arg2** - Verdadeiro se arg1 for menor ou igual a arg2.
- **arg1 -gt arg2** - Verdadeiro se arg1 for maior que arg2.
- **arg1 -ge arg2** - Verdadeiro se arg1 for maior ou igual a arg2.

Exemplos

1. Verificando qual shell.

```
MEU_SHELL="bash"

if [ "$MEU_SHELL" = "bash" ]
    then
        echo "Meu brother...você é o usuário shell zsh!"
fi
```

2. Verificando se um arquivo é legível.

```
#!/bin/bash

# Verificando as prováveis causas de falha:
ls -l /etc/passwd

if [ -r "/etc/passwd" ]; then
    echo "Legivel"
else
    echo "Ihhh Rapaz..: não é um arquivo legível."
fi
```

3. Identificando Sistema Operacional.

```
#!/bin/bash
OS=`uname -s`
if [ "$OS" = "FreeBSD" ]; then
     echo "Este é o FreeBSD"
elif [ "$OS" = "CYGWIN_NT-5.1" ]; then
     echo "Este é o Cygwin"
elif [ "$OS" = "SunOS" ]; then
     echo "Este é Solaris"
elif [ "$OS" = "Darwin" ]; then
     echo "Este é o Mac OSX"
elif [ "$OS" = "Linux" ]; then
     echo "Este é o Linux"
else
     echo "Falha ao identificar este sistema operacional"
fi
```

4. Checando as palavras.

```
shell #!/bin/bash read -p "Dê-me uma palavra:" input echo -en "Isso é" case $input in *[[:digit
:]]*)echo -en "numérico" ;;& *[[:lower:]]*)echo -en "minúsculas" ;;& *[[:upper:]]*)echo -en "maiúsculas
" ;;& *)echo "input." ;; esac
```

Usar itens de looping (for, etc.) para processar a entrada da linha de comando e arquivos ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Crie scripts de shell simples**
  - Executar código condicionalmente (uso de: if, test, [], etc.)
  - **Usar construções de loop (for, etc.) para processar arquivo, entrada de linha de comando**
  - Entradas de script de processo (`$1`, `$2`, etc.)
  - Processar saída de comandos shell em um script
  - Processar códigos de saída de comando shell

loops

Um loop é um bloco de código que repete uma lista de comandos, desde que a condição de controle do loop seja verdadeira.

Sintaxe

Sintaxe para um loop em bash.

```
for variavel in elemento1 elemento2 elemento3
do
```

```
 comandos
done
```

Na Sintaxe acima: - **for**, **in**, **do** e **done'** são palavras-chave; - "**elemento1**", "**elemento2**" ou "**Lista**" contém uma lista de valores ou os valores propriamente ditos. A lista pode ser uma variável que contém várias palavras separadas por espaços. Se a lista estiver faltando na instrução**for', ela receberá o parâmetro posicional que foi passado para o shell;** - variavel** é qualquer nome de variável Bash.

1. Valores estáticos para a lista após a palavra-chave `in`

No exemplo a seguir, a lista de valores (Seg, Ter, Quarta, Qui e Sex) é fornecida diretamente após a palavra-chave `in` no bash for loop.

```
[root@localhost 4linux]# cat for1.sh
i=1
for dia in Seg Ter Qua Qui Sex
do
 echo "Dia da Semana $((i++)) : $dia"
done
```

2. Variável para a lista após a palavra-chave `in`

Em vez de fornecer os valores diretamente no loop for, você pode armazenar os valores em uma variável e usá-la no loop for após a palavra-chave `in`, conforme mostrado no exemplo a seguir.

```
[root@localhost 4linux]# cat for2.sh
i=1
dia_da_semana="Seg Ter Qua Qui Sex"
for dia in $dia_da_semana
do
 echo "dia_da_semana $((i++)) : $dia"
done
```

3. Saída do comando Unix como valores de lista após a palavra-chave `in`

Você pode usar a saída de qualquer comando UNIX/Linux como uma lista de valores para o loop `for`, incluindo o comando entre as aspas simples ‘‘ como mostrado abaixo.

```
[root@localhost 4linux]# cat for3.sh
```

```
i=1
for username in `awk -F: '{print $1}' /etc/passwd`
do
 echo "Nome do Usuario $((i++)) : $username"
done
```

4. Loop através de arquivos e diretórios em um loop `for`

Para percorrer os arquivos e diretórios em um diretório específico, basta fazer o `cd` para esse diretório e fornecer `*` no loop for, conforme mostrado abaixo.

O exemplo a seguir percorrerá todos os arquivos e diretórios em seu diretório inicial:

```
[root@localhost 4linux]# cat for4.sh
i=1
cd ~
for item in *
do
 echo "Item $((i++)) : $item"
done
```

5. Saia do loop `for`

Você pode sair de um loop for usando o comando `break` conforme mostrado abaixo.

```
[root@localhost 4linux]# cat for5.sh
i=1
for dia in Seg Ter Qua Qui Sex
do
 echo "Dia da semana $((i++)) : $dia"
 if [ $i -eq 3 ]; then
   break;
 fi
done
```

6. Continue a partir do topo do loop `for`

Sob certas condições, você pode ignorar o resto dos comandos no loop `for` e continuar o loop do início novamente (para o próximo valor na lista), usando o comando `continue` conforme mostrado abaixo.

O exemplo a seguir adiciona "(FDS)" aos sábados e domingos e "(dia da semana)" aos demais dias.

“‘shell [root@localhost 4linux]# cat for7.sh i=1 for dia in Seg Ter Qua Qui Sex Sab Dom do echo -n “Dia $((i++)) : $dia” if [$i -eq 7 -o $i -eq 8]; then echo " (FDS)" continue; fi echo " (FDS)" done

7

Processar entradas de script ($1, $2 etc.)

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Crie scripts de shell simples**
  - Executar código condicionalmente (uso de: if, test, [], etc.)
  - Usar construções de loop (for, etc.) para processar arquivo, entrada de linha de comando
  - **Entradas de script de processo (`$1`, `$2`, etc.)**
  - Processar saída de comandos shell em um script
  - Processar códigos de saída de comando shell

Entradas Especiais

As chamadas entradas especiais são conhecidas como parâmetros posicionais. Por exemplo, `$0`, `$1`, `$3`, `$4` e assim por diante. `$1` é o primeiro argumento da linha de comando passado para um script de shell. Para você entender melhor, se rodar um script desta forma: **./script.sh filename1 dir1**, vai perceber que:

- `**$0**` - É o nome do proprio script (script.sh);
- `**$1**` - É o primeiro argumento (filename1);
- `**$2**` - É o segundo argumento (dir1);
- `**$9**` - É o nono argumento;
- `**${10}**` - `É o décimo argumento e deve ser colocado entre colchetes após` $9‘;
- `**${11}**` - É o décimo primeiro argumento;

Exemplos

1. Exemplo de parâmetros posicionais

```
#!/bin/bash

script="$0"
primeiro="$1"
segundo="$2"
decimo="${10}"
echo "Nome do script: $script"
echo "O primeiro argumento: $primeiro"
echo "O segundo argumento: $segundo"
echo "O decimo e o decimo primeiro argumento: $decimo e ${11}"
```

Todos os caracteres especiais

- **`$0`** - O nome do arquivo do script atual.
- **`$n`** - Essas variáveis correspondem aos argumentos com os quais um script foi chamado. Aqui, `n` é um número decimal positivo correspondente à posição de um argumento (o primeiro argumento é `$1`, o segundo argumento é `$2` e assim por diante).
- **$#** - O número de argumentos fornecidos a um script.
- **`$\*`** - Todos os argumentos estão entre aspas duplas. Se um script receber dois argumentos, `$\*` é equivalente a `$1 $2`.
- **`$@`** - Todos os argumentos são individualmente aspas duplas. Se um script receber dois argumentos, `$@` é equivalente a `$1 $2`.
- **`$?`** - O status de saída do último comando executado.
- **`$$`** - O número do processo do shell atual. Para scripts de shell, este é o ID do processo sob o qual eles estão executando.
- **`$!`** - O número do processo do último comando em segundo plano.

Funções

Quando se trata de funções, `$1` serve como o primeiro parâmetro da função, e assim por diante.

```
#!/bin/bash
morrer(){
    local m="$1"  # O primeiro argumento
    local e=$2    # o segundo argumento
    echo "$m"
    exit $e
}

# Se não forem exibidos argumentos suficientes, exibir um erro e matar o processo
[ $# -eq 0 ] && morrer "Uso: $0 nomedoarquivo" 1

# O resto do script continua aqui...
echo "Agora podemos comecar a trabalhar no script..."
```

8

Processando saída de comandos de shell em um script

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Crie scripts de shell simples**
  - Executar código condicionalmente (uso de: if, test, [], etc.)
  - Usar construções de loop (for, etc.) para processar arquivo, entrada de linha de comando
  - Entradas de script de processo ($1, $2, etc.)
  - **Processar saída de comandos shell em um script**
  - Processar códigos de saída de comando shell

Saída de Shell

Quando criamos scripts, podemos redirecionar a saída de todas as instruções echo para um arquivo de log sem especificar explicitamente o operador de redirecionamento e o nome do arquivo de log após cada instrução. O comando Bash exec é um poderoso utilitário embutido, que pode ser utilizado para essa finalidade.

Exemplos

Podemos chamar scripts ou outros programas dentro de um script usando `exec` para substituir o processo existente na memória. Isso economiza o número de processos criados e, portanto, os recursos do sistema. Esta implementação é particularmente útil nos casos em que não queremos retornar ao script principal depois que o sub-script ou programa é executado:

```
#! /bin/bash

while true
do
   echo "1. Status dos dscos"
   echo "2. Enviar relatorios da noite"
   read Input
   case "$Input" in
      1) exec df -kh ;;
      2) exec /home/SendReport.sh  ;;
   esac
done
```

Descritores de arquivo e registro em scripts de shell usando o co-

O comando `exec` é uma ferramenta poderosa para manipular descritores de arquivos, criando saída e registro de erros em scripts com o mínimo de alterações. No Linux, por padrão, o descritor de arquivo `0` é `stdin` (a entrada padrão), `1` é `stdout` (a saída padrão) e `2` é `stderr` (o erro padrão) .

Registro em scripts

Podemos abrir, fechar e copiar o `stdout` dinamicamente para realizar as operações de registro. Vamos redirecionar stdout (1) para o arquivo de log:

```
#!/bin/bash
script_log="/tmp/log_`date +%F`.log"
exec 1>>$script_log
echo "Isso sera gravado no arquivo de log em vez do terminal.."
echo "Isso tbm.."
```

Verificamos como podemos gravar a saída padrão em arquivos, agora vamos verificar como também podemos gravar o erro padrão no mesmo arquivo:

```
#!/bin/bash
```

```
script_log="/home/4linux/log_`date +%F`.log"
exec 1>>$script_log
exec 2>&1
date
echo "O comando acima está errado, o erro será registrado no arquivo de log"
date
echo "A saída do comando de data correta também será registrada no arquivo de log,
     incluindo essas declarações de echo"
```

Aqui, copiamos o `stderr` (2) para o `stdout` (1), e o `stdout` já foi alterado para gravar no arquivo de log.

Executando Scripts em um Ambiente Limpo

Podemos redefinir todas as variáveis de ambiente para uma execução limpa usando a opção `-c`:

```
[root@localhost 4linux]# exec -c printenv
```

Como o comando `printenv` lista as variáveis de ambiente, fornecê-lo como um argumento para o comando `exec` aqui imprime uma saída vazia. # Processando códigos de saída de comando shell

Pontos de estudo para o exame

Os candidatos ao exame `RHCSA` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Crie scripts de shell simples**
  - Executar código condicionalmente (uso de: if, test, [], etc.)
  - Usar construções de loop (for, etc.) para processar arquivo, entrada de linha de comando
  - Entradas de script de processo (`$1`, `$2`, etc.)
  - Processar saída de comandos shell em um script
  - **Processar códigos de saída de comando shell**

O que é um código de saída no shell do UNIX ou Linux?

Um código de saída, ou às vezes conhecido como código de retorno, é o código retornado a um processo pai por um executável. Em sistemas POSIX, o código de saída padrão é `0` para

sucesso e qualquer número de `1` a `255` para qualquer outra coisa.

Os códigos de saída podem ser interpretados por scripts de máquina para se adaptar em caso de sucesso ou falha. Se os códigos de saída não forem definidos, o código de saída será o código de saída do último comando de execução.

Como obter o código de saída de um comando

Para obter o código de saída de um comando, digite `echo $?` no prompt de comando. No exemplo a seguir, um arquivo é impresso no terminal usando o comando `cat`.

```
[root@localhost 4linux]# cat file.txt
hello world
echo $?
0
```

O comando foi executado com sucesso. O arquivo existe e não há erros ao ler o arquivo ou gravá-lo no terminal. O código de saída é, portanto, `0`.

```
#!/bin/bash

## Executando o comando:
head /etc/passwd

## Checando se deu tudo certo:
if [ $? -eq 0 ]
then
 echo "Esse script ta rodando fino"
 exit 0
else
 echo "Esse script falhou.." >&2
 exit 1
fi
```

Como definir um código de saída

Para definir um código de saída em um script, use o `exit 0`, em que `0` é o número que você deseja retornar. No exemplo a seguir, um script de shell sai com um `1`. Este arquivo é salvo como `exit.sh`.

```
#!/bin/bash

exit 1
```

A execução deste script mostra que o código de saída está definido corretamente.

```
bash exit.sh
echo $?
1
```

Qual código de saída devo usar? O Projeto de Documentação do Linux tem uma lista de códigos reservados que também oferece conselhos sobre qual código usar em cenários específicos. Estes são os códigos de erro padrão no Linux ou UNIX:

- **1** - Sinal para erros de uma forma geral.
- **2** - Uso indevido de builtins shell (de acordo com a documentação do Bash).
- **126** - Comando invocado não pode ser executado.
- **127** - "Comando não encontrado".
- **128** - Argumento inválido para sair.
- **128+n** - Sinal de erro fatal "n".
- **130** - Script encerrado por Control-C.
- **255*** - Status de saída fora do intervalo.

Como suprimir status de saída

Às vezes, pode haver um requisito para suprimir um status de saída. Pode ser que um comando esteja sendo executado em outro script e qualquer coisa diferente de um status `0` seja indesejável. No exemplo a seguir, um arquivo é impresso no terminal usando `cat`. Este arquivo não existe, então causará um status de saída `1`. Para suprimir a mensagem de erro, qualquer saída de erro padrão é enviada para o `/dev/null` usando `2>/dev/null`.

Se o comando `cat` falhar, uma operação `OR` pode ser usada para fornecer um fallback - `cat file.txt || exit 0`. Nesse caso, um código de saída de `0` é retornado mesmo se houver um erro.

Combinando a saída de supressão de erro e a operação `OR`, o script a seguir retorna um código de status de `0` sem saída, embora o arquivo não exista.

"'shell #!/bin/bash

cat 'naoexiste.txt' 2>/dev/null || exit 0 "'# Inicializar, reinicializar e desligar um sistema normalmente

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - **Inicialize, reinicie e desligue um sistema normalmente**
  - Inicialize sistemas em diferentes alvos manualmente
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - Identifique processos intensivos de CPU/memória e elimine processos
  - Ajuste a programação de processos
  - Gerencie perfis de ajuste
  - Localize e interprete os arquivos de registro do sistema e diários
  - Preserve diários do sistema
  - Inicie, pare e verifique o status dos serviços de rede
  - Transfira arquivos com segurança entre sistemas

Introdução

O desligamento refere-se ao processo de parar e desligar um computador ou servidor. Isso envolve cortar a energia dos principais componentes do sistema, usando um processo controlado. Os aplicativos são fechados, os processos e protocolos ativos são salvos no disco rígido, os drivers de dispositivo são removidos e as configurações do usuário são salvas no processo. Os sistemas operacionais Linux podem ser facilmente interrompidos, desligados e reiniciados usando o comando shutdown e suas várias opções. Os comandos de desligamento do Linux são inseridos no terminal Linux, que é iniciado usando o atalho de teclado `[Ctrl]` + `[Alt]` + `[T]`. Você pode então fechar a janela do terminal com o atalho `[Ctrl]` + `[D]`.

Tabela de comparação de comandos de gerenciamento de energia

Comando antigo	Comando novo	Descricao
halt	systemctl halt	Para o sistema.
poweroff	systemctl poweroff	Desliga o sistema.
reboot	systemctl reboot	Reinicia o sistema.
pm-suspend	systemctl suspend	Suspende o sistema.
pm-hibernate	systemctl hibernate	Hiberna o sistema.
pm-suspend-hybrid	systemctl hybrid-sleep	Hiberna e suspende o sistema.

Gestão Básica

Para reinicializar o sistema, escolha um comando entre estes:

```
[root@localhost 4linux]# reboot
[root@localhost 4linux]# systemctl reboot
[root@localhost 4linux]# shutdown -r now
[root@localhost 4linux]# init 6
[root@localhost 4linux]# telinit 6
```

Para desligar o sistema, escolha um comando entre estes:

```
[root@localhost 4linux]# halt
[root@localhost 4linux]# systemctl halt
[root@localhost 4linux]# shutdown -h now
[root@localhost 4linux]# init 0
[root@localhost 4linux]# telinit 0
```

Para desligar o sistema, escolha um comando entre estes:

```
[root@localhost 4linux]# poweroff
[root@localhost 4linux]# systemctl poweroff
```

Gestão Avançada

Para suspender o sistema, digite:

```
[root@localhost 4linux]# systemctl suspend
```

Para colocar o sistema em hibernação, digite:

```
[root@localhost 4linux]# systemctl hibernate
```

Para colocar o sistema em hibernação e suspendê-lo, digite:

```
[root@localhost 4linux]# systemctl hybrid-sleep
```

Os comandos essenciais de desligamento do Linux

Ao desligar ou reiniciar o Linux através do terminal, o comando `shutdown` é essencial. Você pode adicionar uma opção seguida por uma especificação de tempo e uma mensagem. A sintaxe do comando de desligamento do Linux é a seguinte:

```
shutdown [OPTION] [TIME] [MESSAGE]
```

Há pelo menos uma alternativa para cada comando listado aqui que produz o mesmo resultado.

Mais algumas opções:

```
[root@localhost 4linux]# shutdown -h
```

```
[root@localhost 4linux]# shutdown -h 0
```

```
[root@localhost 4linux]# shutdown -r 0
```

```
[root@localhost 4linux]# shutdown +20
```

```
[root@localhost 4linux]# shutdown -r +20
```

```
[root@localhost 4linux]# shutdown -h 17:30
```

```
[root@localhost 4linux]# shutdown -r 17:30
```

Desligamento do Linux - comandos adicionais

Além dos comandos de desligamento do Linux mencionados anteriormente, há vários outros comandos e opções para parar, desligar e reiniciar os sistemas operacionais Linux. Geralmente,

eles também podem ser combinados com comandos para desligamentos programados do Linux .

> Há uma diferença entre "parar um sistema" e "desligar um sistema". Quando você o interrompe, todos os processadores (CPUs) são interrompidos, mas quando você o desliga, ele também é cortado da fonte de alimentação principal. Geralmente, o termo "desligamento" é entendido como a parada e o desligamento de um sistema.

Comando para desligar o Linux

```
[root@localhost 4linux]# shutdown -P
```

Este comando indica explicitamente que o sistema será encerrado e a fonte de alimentação principal será cortada.

Comando para configurar uma mensagem

```
[root@localhost 4linux]# shutdown 'ESCREVA SUA MENSAGEM AQUI!!'
```

Uma mensagem é uma informação exibida na tela dos usuários do sistema operacional. Por exemplo, um administrador pode usar uma mensagem de parede para informar aos usuários que o sistema está sendo encerrado.

Comando para cancelar desligamentos ou reinicializações programadas

```
[root@localhost 4linux]# shutdown -c
```

Usando este comando, você pode cancelar um desligamento ou reinicialização agendada. Isso requer que o processo ainda não tenha iniciado.

Usando comandos simples no terminal Linux, você pode parar, desligar e reiniciar seu sistema operacional. Como alternativa para inserir os comandos de desligamento do Linux diretamente, você pode instalar uma interface gráfica de usuário usando software, como o desligamento do programa, que é especialmente adequado para a distribuição do Linux.

9

Inicialize sistemas em diferentes alvos manualmente

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - **Inicialize sistemas em diferentes alvos manualmente**
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - Identifique processos intensivos de CPU/memória e elimine processos
  - Ajuste a programação de processos
  - Gerencie perfis de ajuste
  - Localize e interprete os arquivos de registro do sistema e diários
  - Preserve diários do sistema
  - Inicie, pare e verifique o status dos serviços de rede
  - Transfira arquivos com segurança entre sistemas

Inicializando sistemas

As versões anteriores do Linux eram distribuídas com `SysV init` ou `Upstart` e implementavam um conjunto predefinido de níveis de execução que representavam modos específicos de operação.

Esses níveis de execução eram numerados de `0` a `6` e definidos por uma seleção de serviços do sistema a serem executados quando um determinado nível era ativado pelo administrador do sistema.

Com o tempo, o conceito de níveis de execução foi substituído por destinos do `systemd`. Os destinos do `Systemd` são representados por unidades de destino. As unidades de destino terminam com a extensão de arquivo `.target` e seu único propósito é agrupar outras unidades do `systemd` por meio de uma cadeia de dependências. Por exemplo, a unidade `graphical.target`, que é usada para iniciar uma sessão gráfica, inicia serviços do sistema como o GNOME Display Manager (gdm.service) ou Accounts Service (accounts-daemon.servico), e também ativa a unidade `multi-user.target`.

Da mesma forma, a unidade `multi-user.target` inicia outros serviços essenciais do sistema, como NetworkManager (NetworkManager.service) ou D-Bus (dbus.service) e ativa outra unidade de destino chamada `basic.target`. O Red Hat Enterprise Linux 8 é distribuído com vários destinos predefinidos que são mais ou menos semelhantes ao conjunto padrão de níveis de execução das versões anteriores deste sistema. Por motivos de compatibilidade, ele também fornece aliases para esses destinos que os mapeiam diretamente para níveis de execução `SysV`.

Nível de execução	Unidades Alvo	Descrição
0	runlevel0.target, poweroff.target	Alterar seu sistema para o nível de execução 0 desligará o sistema e desligará seu servidor / desktop.
1	runlevel1.target, rescue.target	Também conhecido como modo único, o nível de execução de resgate é usado para solução de problemas do sistema e várias tarefas de administração do sistema.
2	runlevel2.target, multi-user.target	Nível de execução definido pelo usuário. Por padrão, é idêntico ao nível de execução 3.

Nível de execução	Unidades Alvo	Descrição
3	runlevel3.target, multi-user.target	Este é um nível de execução multiusuário e não gráfico. Vários usuários podem fazer login por meio de consoles / terminais locais ou acesso remoto à rede.
4	runlevel4.target, multi-user.target	Nível de execução definido pelo usuário. Por padrão, é idêntico ao nível de execução 3.
5	runlevel5.target, graphical.target	Nível de execução gráfico multiusuário. Vários usuários podem fazer login por meio de consoles / terminais locais ou acesso remoto à rede.
6	runlevel6.target, reboot.target	Alterar seu sistema para este nível de execução irá reinicializar seu sistema.

Comando antigo	Comando novo	Descrição
runlevel	systemctl list-units –type target	ALista as unidades de destino atualmente carregadas.
telinit runlevel	systemctl isolate name.target	Altera o alvo atual.

Mudando o alvo atual

Para mudar para uma unidade de destino diferente na sessão atual, digite o seguinte comando:

```
[root@localhost 4linux]# systemctl isolate name.target
```

Este comando inicia a unidade alvo denominada nome e todas as unidades dependentes, e interrompe imediatamente todas as outras.

Mudando o alvo atual

Para desligar a interface gráfica do usuário e mudar para a unidade `multi-user.target` na sessão atual, execute o seguinte comando como root:

```
[root@localhost 4linux]# systemctl isolate multi-user.target
```

Mudando para o rescue mode

O modo de "recuperação" fornece um ambiente conveniente para um único usuário e permite que você repare seu sistema em situações em que ele não consegue concluir um processo de inicialização normal. No modo de recuperação, o sistema tenta montar todos os arquivos locais e iniciar alguns serviços importantes do sistema, mas não ativa as interfaces de rede nem permite que mais usuários façam login no sistema ao mesmo tempo. O modo de recuperação é equivalente ao modo de usuário único e requer a senha do root.

Para alterar o destino atual e entrar no modo de recuperação na sessão atual, digite o seguinte comando:

```
[root@localhost 4linux]# systemctl rescue
```

Esse comando é semelhante ao `systemctl isolate rescue.target`, mas também envia uma mensagem informativa a todos os usuários que estão atualmente logados no sistema. Para evitar que o `systemd` envie esta mensagem, execute este comando com a opção de linha de comando `--no-wall`:

```
[root@localhost 4linux]# systemctl --no-wall rescue
```

Mudando para o modo de emergência

O modo de emergência fornece o ambiente mínimo possível e permite que você repare seu sistema mesmo em situações em que ele não consegue entrar no modo de resgate. No modo de emergência, o sistema monta os arquivos raiz apenas para leitura, não tenta montar nenhum

outro sistema de arquivos local, não ativa interfaces de rede e inicia apenas alguns serviços essenciais.

Para alterar o alvo atual e entrar no modo de emergência, digite o seguinte em um prompt de shell como root:

```
[root@localhost 4linux]# systemctl Emergency
```

Este comando é semelhante a `systemctl isolate Emergency.target`, mas também envia uma mensagem informativa a todos os usuários que estão atualmente logados no sistema.

10

Interromper o processo de inicialização, a fim de obter acesso a um sistema.

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - Inicialize sistemas em diferentes alvos manualmente
  - **Interrompa o processo de inicialização para obter acesso a um sistema**
  - Identifique processos intensivos de CPU/memória e elimine processos
  - Ajuste a programação de processos
  - Gerencie perfis de ajuste
  - Localize e interprete os arquivos de registro do sistema e diários
  - Preserve diários do sistema
  - Inicie, pare e verifique o status dos serviços de rede
  - Transfira arquivos com segurança entre sistemas

Redefinindo a senha de root usando rd.break

No início do processo de inicialização, no menu GRUB2, digite a chave `e` para editar. Em seguida, vá para a linha do kernel (a linha que começa com linux16) e adicione as seguintes instruções no final:

```
rd.break enforcing=0
```

O `rd.break` pede uma pausa no estágio inicial do processo de inicialização. O `enforcing=0` coloca o sistema SELinux em modo Permissive (discutido posteriormente). Pressione `Ctrl+ x` para retomar o processo de inicialização. Em seguida, monte a partição `/sysroot` como read/write:

```
switch_root:/# mount -o remount,rw /sysroot
```

Execute o comando `chroot` na partição `/sysroot`:

```
switch_root:/# chroot /sysroot
```

Mude a senha do root:

```
sh-4.2# passwd root
Changing password for user root.
New passwd: mypassword
Retype new password: mypassword
passwd: all authentication token updated successfully.
sh-4.2# exit
exit
switch_root:/# exit
logout
```

Conecte-se ao seu servidor no console (não reinicie agora!) com o usuário root e a nova senha:

```
[  OK  ] Started Network Manager Script Dispatcher Service.
[  OK  ] Started Crash recovery kernel arming.
[  OK  ] Reached target Multi-User System.

CentOS Linux 7 (Core)
Kernel 3.10.0-229.14.1.el7.x86_64 on an x86_64
```

```
vm login: root
Password: mypassword
```

Em seguida, digite:

`shell [root@localhost 4linux]# restorecon /etc/shadow [root@localhost 4linux]# reboot`# Identifique os processos intensivos de CPU/memória e elimine

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - Inicialize sistemas em diferentes alvos manualmente
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - **Identifique processos intensivos de CPU/memória e elimine processos**
  - Ajuste a programação de processos
  - Gerencie perfis de ajuste
  - Localize e interprete os arquivos de registro do sistema e diários
  - Preserve diários do sistema
  - Inicie, pare e verifique o status dos serviços de rede
  - Transfira arquivos com segurança entre sistemas

Introdução

Identificar problemas em um sistema é muito importante e pode evitar dores de cabeça no futuro. Você pode usar comandos como `ps`, `top`, `kill` e `renice` para gerenciar e monitorar processos em seu sistema Linux.

O comando ps

Este comando é usado para gerar uma análise instantânea de seus processos atuais. O `ps` exibe informações sobre vários processos ativos. Existem muitos sinalizadores que você pode usar junto com o `ps`, abaixo vamos listar alguns exemplos. Para ver cada processo no sistema, use a sintaxe padrão:

```
[root@localhost 4linux]# ps -ely
```

Para ver cada processo no sistema usando a sintaxe ‘ps aux:

```
[root@localhost 4linux]# ps aux
```

Processos em árvore:

```
[root@localhost 4linux]# ps -e --forest
```

Comando `top`

O comando `top` exibe uma visão dinâmica em tempo real das tarefas em execução em seu sistema. Ele também exibe informações resumidas do sistema quanto ao uso da CPU, uso da memória, tempo de atividade e muito mais.

Para usar `top`, basta digitar `top` em seu terminal.

```
[root@localhost 4linux]# top
```

Comando `kill`

O comando `kill` pode ser útil quando você precisa interromper um processo específico. Você pode chamar um processo usando a identificação ou o nome do processo. Existem 2 ferramentas com as quais você precisa estar familiarizado, `kill` e `killall`.

Se você quiser usar `kill` para encerrar um processo com base no id do processo, você deve usar o seguinte script:

```
[root@localhost 4linux]# kill 2014
```

O número 2014 você substituiria pelo id do processo do aplicativo que deseja encerrar. Você também pode matar um processo com base no nome. Por exemplo, digamos que quiséssemos

interromper o processo denominado `firefox`. Nós usaríamos o seguinte:

```
[root@localhost 4linux]# killall firefox
```

Ambas as ferramentas também oferecem vários sinalizadores para lidar com determinados processos de determinadas maneiras. Por exemplo, o sinalizador `-9` pode ser adicionado para `kill` ou `killall`. Assim, matará o processo imediatamente.

Comando `renice`

O comando `renice` permite priorizar seus processos. Isso é útil quando você deseja garantir que um determinado processo seja concluído ou executado em comparação com outros processos. O intervalo das prioridades é de `-20` a `19`. Quanto menor for o número, maior será a prioridade. No entanto, apenas o root pode definir a prioridade para menos que `0`.

Para usar o `renice`, você deve fazer o seguinte:

```
[root@localhost 4linux]# renice -20 process_id
```

O comando acima define o processo com a prioridade mais alta de `-20`. Você substituiria `process_id` pelo id real do seu processo.

```
[root@localhost 4linux]# renice -20 1
```

Relatórios do sistema

Para exibir detalhes sobre as atividades IO, digite:

```
[root@localhost 4linux]# iostat
```

Para mostrar as atividades da placa de rede, digite:

```
[root@localhost 4linux]# netstat -i
```

Para exibir atividades de soquete, digite:

```
[root@localhost 4linux]# netstat -a
```

Para obter detalhes sobre as atividades da memória virtual (memória, troca, fila de execução, uso da CPU etc.) a cada 5 segundos, digite:

```
[root@localhost 4linux]# vmstat 5
```

Para obter um relatório completo de uma atividade do servidor, digite:

`shell [root@localhost 4linux]# sar -A`# Ajustar a programação de processos

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - Inicialize sistemas em diferentes alvos manualmente
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - Identifique processos intensivos de CPU/memória e elimine processos
  - **Ajuste a programação de processos**
  - Gerencie perfis de ajuste
  - Localize e interprete os arquivos de registro do sistema e diários
  - Preserve diários do sistema
  - Inicie, pare e verifique o status dos serviços de rede
  - Transfira arquivos com segurança entre sistemas

Introdução

Ao usar o sistema operacional Linux, vários programas aguardam para serem convertidos em um processo de execução. Quando um programa se transforma em um processo, muitos atributos são configurados e podem ser manipulados. Para isso, o sistema Linux fornece uma ferramenta útil para definir ou buscar atributos em tempo real de um processo.

O comando `chrt` é parte de um utilitário Linux de baixo nível que não é usado apenas para

definir atributos em tempo de execução, mas também para alterar a política de agendamento de um processo e definir sua prioridade. Ele usa o PID existente de qualquer programa em espera para definir e recuperar uma programação de atributos em tempo real. Simplesmente, o escalonador decide qual processo é executado pela CPU primeiro quando o comando `chrt` muda sua prioridade.

Algoritmos de escalonamento:

Existem cinco opções de política de agendamento:

- **SCHED_FIFO** - Esta política usa o algoritmo **First In_First Out**. É um processo em tempo real que suporta apenas uma fila de ordem dos processos.
- **SCHED_BATCH** - Esta política usa o algoritmo de processos em lote.
- **SCHED_RR** - Esta política usa o algoritmo do processo Round Robin.
- **SCHED_IDLE** - Esta política usada para executar trabalhos de E/S com menos prioridade.
- **SCHED_OTHER** - Esta política usa o algoritmo de agendamento Linux-time_sharing padrão.

Sintaxe do comando Chrt

A sintaxe do utilitário de comando `chrt` é:

```
[4linux@localhost ~]$ chrt [ opções ] -p [ prioridade ] pid
```

Comando Chrt com Opções

Para verificar a política de agendamento atual e a prioridade de qualquer programa em execução, encontre seu pid primeiro usando o comando `chrt`.

Por exemplo, para obter o pid de `top`, execute o comando fornecido:

```
[4linux@localhost ~]$ pidof -s top
```

Política Atual e Prioridade de Processo

Para obter o processo de agendamento atual e a prioridade do programa, use `pid`:

```
[4linux@localhost ~]$ chrt -p 3050
```

Prioridade mínima/máxima válida do algoritmo

Para obter as prioridades mínimas e máximas das políticas de agendamento, use a opção `-m`.

```
[4linux@localhost ~]$ chrt -m
```

Alterar Política de Agendamento SCHED_FIFO com Prioridade

Para alterar a política de agendamento de um processo e definir seu nível de prioridade, execute a opção abaixo mencionada com o comando `chrt`.

Por exemplo, a programação atual do programa é `Sched_Batch` e queremos alterá-la para `Sched_Fifo`.

```
[4linux@localhost ~]$ chrt -f -p 15 3050
```

Alterar política de agendamento SCHED_IDLE com prioridade

Definimos a política de agendamento do `top` para `SCHED_FIFO`, agora, para alterá-lo para `SCHED_IDLE` , use o comando:

```
[4linux@localhost ~]$ chrt -i -p 0 3050
```

Ajuda de exibição

Para obter ajuda sobre o comando `chrt`, use `--help` no terminal:

`shell [4linux@localhost ~]$ chrt --help`# Gerenciar perfis de ajuste

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - Inicialize sistemas em diferentes alvos manualmente
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - Identifique processos intensivos de CPU/memória e elimine processos
  - Ajuste a programação de processos
  - **Gerencie perfis de ajuste**
  - Localize e interprete os arquivos de registro do sistema e diários
  - Preserve diários do sistema
  - Inicie, pare e verifique o status dos serviços de rede
  - Transfira arquivos com segurança entre sistemas

Daemon tuned

O daemon tuned é um daemon poderoso que ajusta automatica e dinamicamente o desempenho do servidor Linux com base nas informações coletadas do monitoramento do sistema e de seus componentes subjacentes, a fim de fornecer as características de desempenho solicitadas.

Para instalar o tuned (se ainda não estiver):

```
[root@localhost 4linux]# yum install tuned
```

Perfis de ajuste

Um perfil de ajuste consiste em uma lista de alterações do sistema correspondentes a um requisito específico. Para obter a lista dos perfis de ajuste disponíveis, digite:

```
[root@localhost 4linux]# tuned-adm list
Available profiles:
- accelerator-performance     - Throughput performance based tuning with disabled higher
     latency STOP states
- balanced                    - General non-specialized tuned profile
- desktop                     - Optimize for the desktop use-case
- hpc-compute                 - Optimize for HPC compute workloads
- intel-sst                   - Configure for Intel Speed Select Base Frequency
- latency-performance         - Optimize for deterministic performance at the cost of
     increased power consumption
```

```
- network-latency             - Optimize for deterministic performance at the cost of
      increased power consumption, focused on low latency network performance
- network-throughput          - Optimize for streaming network throughput, generally only
      necessary on older CPUs or 40G+ networks
- optimize-serial-console     - Optimize for serial console use.
- powersave                   - Optimize for low power consumption
- throughput-performance      - Broadly applicable tuning that provides excellent
      performance across a variety of common server workloads
- virtual-guest               - Optimize for running inside a virtual guest
- virtual-host                - Optimize for running KVM guests
Current active profile: virtual-guest
```

Para obter apenas o perfil ativo, digite:

```
[root@localhost 4linux]# tuned-adm active
Current active profile: virtual-guest
```

Para obter o perfil de ajuste recomendado em sua configuração atual, digite:

```
[root@localhost 4linux]# tuned-adm recommend
virtual-guest
```

Para aplicar um perfil de ajuste diferente:

```
[root@localhost 4linux]# tuned-adm profile throughput-performance
```

Criação de Perfil

Em alguns casos, como não é aconselhável alterar um perfil existente, criar um novo perfil pode ser a melhor opção.

Vá para o diretório de perfis de ajuste:

```
[root@localhost 4linux]# cd /usr/lib/tuned/
```

Observação: como alternativa, você pode criar o diretório `/etc/tuned` se ele ainda não existir. Então, crie um novo diretório com o nome do perfil escolhido:

```
[root@localhost 4linux]# mkdir sas--performance
```

No caso do SAS, a opção mais fácil é fazer com que o novo perfil seja herdado do perfil de desempenho de rendimento. Crie um novo arquivo no diretório `sas-performance` chamado `tuned.conf` e cole as seguintes linhas:

```
[main]
include=throughput--performance
```

Todas as alterações feitas além da instrução de inclusão substituirão o perfil de desempenho de rendimento.

Em seguida, ative o novo perfil:

```
[root@localhost 4linux]# tuned--adm profile sas--performance
```

Tunel Dinamico

Tuned também pode ajustar dinamicamente sua configuração. Isso é feito por meio do arquivo `/etc/tuned/tuned-main.conf`. Neste arquivo, você precisa atribuir um valor de 1 ao parâmetro `dynamic_tuning`:

dynamic_tuning = 1 Em seguida, você precisa reiniciar o serviço ajustado:

```
[root@localhost 4linux]# systemctl restart tuned
```

Para ter uma ideia da nova configuração, verifique o arquivo `/var/log/tuned/tuned.log`:

```
shell [root@localhost 4linux]# tail -f /var/log/tuned/tuned.log 2021-07-11 09:49:34,850 INFO tuned
 .daemon.application: dynamic tuning is enabled (can be overriden in plugins)2021-07-11 09:49:34,850
  INFO tuned.daemon.daemon: using sleep interval of 1 second(s)2021-07-11 09:49:34,850 INFO tuned
 .daemon.daemon: dynamic tuning is enabled (can be overriden by plugins)2021-07-11 09:49:34,850
  INFO tuned.daemon.daemon: using update interval of 10 second(s)(10 times of the sleep interval
 )2021-07-11 09:49:34,851 INFO tuned.profiles.loader: loading profile: virtual-guest 2021-07-11
  09:49:34,855 INFO tuned.daemon.controller: starting controller 2021-07-11 09:49:34,855 INFO tuned
```

`.daemon.daemon: starting tuning`# Localize e interprete os arquivos de registro do sistema e diários

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - Inicialize sistemas em diferentes alvos manualmente
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - Identifique processos intensivos de CPU/memória e elimine processos
  - Ajuste a programação de processos
  - Gerencie perfis de ajuste
  - **Localize e interprete os arquivos de registro do sistema e diários**
  - Preserve diários do sistema
  - Inicie, pare e verifique o status dos serviços de rede
  - Transfira arquivos com segurança entre sistemas

Introdução

Algumas das vantagens mais interessantes do `systemd` são aquelas envolvidas com o registro de processos e do sistema. Ao usar outros sistemas, os registros ficam geralmente dispersos, sendo manuseados por daemons e processos diferentes. Isso torna a interpretação deles bastante difícil quando englobam vários aplicativos. O systemd tenta resolver esses problemas fornecendo uma solução de gerenciamento centralizada para registrar todos os processos do kernel e do userland. O sistema que coleta e gerencia esses registros é conhecido como journal (**Journal**).

O **Journal** é implementado com o daemon `journald`, que manuseia todas as mensagens produzidas pelo kernel, initrd, serviços etc. Neste guia, vamos discutir como utilizar o utilitário journalctl, que pode servir para acessar e manipular os dados mantidos dentro do **Journal**.

Ideal geral

Um dos motivos da existência do **Journal** do `systemd` é o de centralizar o gerenciamento de registros independentemente de onde as mensagens estão sendo originadas. Como uma grande parte do processo de inicialização do sistema e gerenciamento de serviços é manuseada pelo processo `systemd`, faz sentido padronizar a maneira como os registros são coletados e acessados.

O daemon journald coleta dados de todas as fontes disponíveis e os armazena em um formato binário para uma manipulação fácil e dinâmica.

Isso nos dá várias vantagens significativas. Ao interagir com os dados usando um único utilitário, os administradores são capazes de exibir dinamicamente dados de registro de acordo com suas necessidades. Isso pode ser algo simples como visualizar os dados de inicialização de três inicializações de sistema atrás ou combinar as entradas de registro sequencialmente de dois serviços relacionados para consertar um problema de comunicação.

Armazenar os dados de registro em um formato binário também faz com que eles possam ser exibidos em formatos de saída arbitrários, dependendo da sua necessidade naquele momento. Por exemplo, para o gerenciamento de registros **Journal**s, você pode estar acostumado a visualizar os registros no formato syslog padrão. No entanto, se você quiser representar interrupções de serviço em gráficos mais tarde, é possível gerar um objeto JSON para cada entrada para que seja consumível pelo seu serviço de criação de gráficos. Como os dados não são escritos no disco em texto simples, nenhuma conversão é necessária quando houver a demanda por um formato diferente.

O **Journal** do `systemd` pode ser usado com uma implementação do syslog existente ou, ainda, pode substituir a funcionalidade syslog, dependendo das suas necessidades. Embora o **Journal** do systemd atenda a maioria das necessidades de registro de administrador, ele também pode complementar mecanismos de registro existentes. Por exemplo, pode ser que você tenha um servidor syslog centralizado que usa para compilar dados de vários servidores, mas também queira intercalar os registros de vários serviços em um único sistema com o **Journal** do systemd. É possível fazer as duas coisas combinando essas tecnologias.

Configurando o horário do sistema

Um dos benefícios do uso de um **Journal** binário para registro é a capacidade de visualizar registros em UTC ou em seu horário local à vontade. Por padrão, o `systemd` exibirá resultados no horário local.

Por conta disso, antes de começarmos com o **Journal**, vamos garantir que o fuso horário esteja configurado corretamente. O pacote do `systemd` vem com uma ferramenta chamada `timedatectl` que pode ajudar com isso.

Primeiramente, consulte quais fusos horários estão disponíveis com a opção `list-timezones`:

```
[root@localhost 4linux]# timedatectl list-timezones
```

Isso listará os fusos horários disponíveis no seu sistema. Depois de encontrar aquele que corresponde ao local do seu servidor, defina-o usando a opção `set-timezone`:

```
[root@localhost 4linux]# timedatectl set-timezone zone
```

Para garantir que sua máquina esteja usando o horário correto neste momento, use o comando `timedatectl` sozinho, ou com a opção status. O resultado será o mesmo:

```
[root@localhost 4linux]# timedatectl status
```

```
      Local time: Thu 2021-02-05 14:08:06 EST
  Universal time: Thu 2021-02-05 19:08:06 UTC
        RTC time: Thu 2021-02-05 19:08:06
       Time zone: America/New_York (EST, -0500)
     NTP enabled: no
NTP synchronized: no
 RTC in local TZ: no
      DST active: n/a
```

A primeira linha deve exibir o horário correto.

Visualização básica de registros

Para ver os registros que o daemon do `journald` coletou, use o comando `journalctl`.

Quando usado sozinho, todas as entradas no **Journal** que estão no sistema serão exibidas dentro de um pager (geralmente `less` (menos)) para você navegar. As entradas mais antigas estarão no topo:

```
[root@localhost 4linux]# journalctl
```

```
-- Logs begin at Tue 2021-02-03 21:48:52 UTC, end at Tue 2021-02-03 22:29:38 UTC. --
Feb 03 21:48:52 localhost.localdomain systemd-journal[243]: Runtime journal is using 6.2M (
     max allowed 49.
Feb 03 21:48:52 localhost.localdomain systemd-journal[243]: Runtime journal is using 6.2M (
     max allowed 49.
Feb 03 21:48:52 localhost.localdomain systemd-journald[139]: Received SIGTERM from PID 1 (
     systemd).
Feb 03 21:48:52 localhost.localdomain kernel: audit: type=1404 audit(1423000132.274:2):
     enforcing=1 old_en
Feb 03 21:48:52 localhost.localdomain kernel: SELinux: 2048 avtab hash slots, 104131 rules.
```

```
Feb 03 21:48:52 localhost.localdomain kernel: SELinux: 2048 avtab hash slots, 104131 rules.
Feb 03 21:48:52 localhost.localdomain kernel: input: ImExPS/2 Generic Explorer Mouse as /
     devices/platform/
Feb 03 21:48:52 localhost.localdomain kernel: SELinux:  8 users, 102 roles, 4976 types, 294
      bools, 1 sens,
Feb 03 21:48:52 localhost.localdomain kernel: SELinux:  83 classes, 104131 rules

. . .
```

É provável que você tenha páginas e páginas de dados para percorrer, que podem ser dezenas ou centenas de milhares de linhas se o `systemd` estiver em seu sistema há bastante tempo. Isso demonstra a quantidade de dados que está disponível no banco de dados do **Journal**.

O formato será conhecido por aqueles que estão acostumados com o registro padrão do `syslog`. No entanto, esse processo coleta dados de mais fontes do que as implementações tradicionais do syslog são capazes de fazer. Ele inclui registros do processo de inicialização inicial, do kernel, do initrd e do erro padrão de aplicativo e mais. Todos eles estão disponíveis no **Journal**.

Note que todos os carimbos de data/hora estão sendo exibidos no horário local. Isso está disponível para toda entrada de registro agora que temos nosso horário local configurado corretamente em nosso sistema. Todos os registros são exibidos usando essas novas informações.

Se quiser exibir os carimbos de data/hora em UTC, use o sinalizador `--utc`:

```
[root@localhost 4linux]# journalctl --utc
```

Filtrar os Logs pela hora

Embora ter acesso a uma grande coleção de dados seja definitivamente útil, grandes quantidades de informações podem ser difíceis ou impossíveis de ser inspecionadas e processadas mentalmente. Por conta disso, uma das características mais importantes do `journalctl` são suas opções de filtragem.

Exibir registros da inicialização atual

A opção mais básica existente, que pode ser usada diariamente, é a flag `-b`. Ele irá mostrar todas as entradas do **Journal** que foram coletadas desde a reinicialização mais recente.

```
[root@localhost 4linux]# journalctl -b
```

Isso ajudará você a identificar e gerenciar informações pertinentes ao seu ambiente atual.

Nos casos em que você não estiver usando esse recurso e estiver exibindo mais de um dia de inicializações, verá que o `journalctl` insere uma linha que se parece com esta, sempre que o sistema for desligado:

```
. . .
-- Reboot --
. . .
```

Isso pode ser usado para ajudar a separar as informações em sessões de inicialização de maneira lógica.

Intervalos de tempo

Embora a visualização de entradas de registro com base na inicialização seja incrivelmente útil, muitas vezes é desejável solicitar intervalos de tempo que não se alinham com as inicializações do sistema. Isso pode ser especialmente válido ao lidar com servidores de execução prolongada com um tempo de atividade significativo.

É possível filtrar por limites arbitrários de data/hora usando as opções `--since` e `--until`, que restringem as entradas exibidas a aquelas que sucedem ou antecedem um determinado tempo, respectivamente.

Os valores de data/hora podem ser usados em uma variedade de formatos. Para valores de data/hora absolutos, use o seguinte formato:

```
YYYY-MM-DD HH:MM:SS
```

Por exemplo, podemos ver todas as entradas desde 10 de janeiro de 2021 às 5:15 PM digitando:

```
[root@localhost 4linux]# journalctl --since "2021-01-10 17:15:00"
```

Se algum componente do formato acima for deixado de fora, o padrão será aplicado. Por exemplo, se a data for omitida, a data atual será empregada. Se o componente de hora estiver faltando, "00:00:00" (meia-noite) será utilizado. O campo de segundos também pode ser deixado de fora e o padrão "00" é empregado:

```
[root@localhost 4linux]# journalctl --since "2021-01-10" --until "2021-01-11 03:00"
```

O **Journal** também compreende alguns valores relativos e seus atalhos nomeados. Por exemplo, é possível usar as palavras "yesterday" (ontem), "today" (hoje), "tomorrow" (amanhã) ou "now" (agora). É possível criar datas/horas relativas prefixando "-" ou "+" a um valor numerado ou usando palavras como "ago" (atrás) em uma construção de sentenças.

Para obter os dados de ontem, utilize o script:

```
[root@localhost 4linux]# journalctl --since yesterday
```

Se tiver recebido relatórios de uma interrupção de serviço iniciada às 9:00 AM que durou até uma hora atrás, você pode digitar:

```
[root@localhost 4linux]# journalctl --since 09:00 --until "1 hour ago"
```

Como se vê, é relativamente fácil definir intervalos flexíveis de tempo para filtrar as entradas que você deseja visualizar.

Filtrar por interesse de mensagens

Aprendemos acima algumas maneiras de filtrar os dados do **Journal** usando restrições de tempo. Nesta seção, vamos discutir como filtrar com base em qual serviço ou componente você está interessado. O **Journal** do systemd oferece diversas maneiras de fazer isso.

Por unidade

Talvez a maneira mais útil de filtrar seja pela unidade na qual você está interessado. Podemos usar a opção `-u` para filtrar dessa maneira.

Por exemplo, para ver todos os registros de uma unidade Crond em nosso sistema, digitamos:

```
[root@localhost 4linux]# journalctl -u crond.service
```

Normalmente, também seria interessante filtrar pela data/hora para exibir as linhas nas quais

você está interessado. Por exemplo, para verificar como o serviço está funcionando hoje, digite:

```
[root@localhost 4linux]# journalctl -u crond.service --since today
```

Esse tipo de foco torna-se extremamente útil quando se aproveita da capacidade do **Journal** de intercalar os registros de várias unidades. Por exemplo, se seu processo Nginx estiver conectado a uma unidade PHP-FPM para processar conteúdo dinâmico, é possível fundir as entradas de ambos em ordem cronológica especificando ambas as unidades:

```
[root@localhost 4linux]# journalctl -u nginx.service -u php-fpm.service --since today
```

Isso pode facilitar a detecção de interações entre diferentes programas e sistemas de depuração, em vez de processos individuais.

Por processo, usuário ou ID de grupo

Alguns serviços geram uma variedade de processos filhos para funcionar. Se você tiver pesquisado o PID exato do processo em que está interessado, também é possível filtrar por ele.

Para fazer isso, especificamos o campo `_PID`. Por exemplo, se o PID em que estivermos interessados for `1`, digitamos:

```
[root@localhost 4linux]# journalctl _PID=1
```

Em outros momentos, pode ser desejável exibir todas as entradas registradas a partir de um usuário ou grupo específico. Isso pode ser feito com os filtros `_UID` ou `_GID`. Por exemplo, se seu servidor Web estiver sendo executado sob o usuário www-data, é possível encontrar o ID do usuário digitando:

```
[root@localhost 4linux]# id -u joatham
```

Depois disso, use o ID retornado para filtrar os resultados do **Journal**:

```
[root@localhost 4linux]# journalctl _UID=1000 --since today
```

O **Journal** do systemd possui muitos campos que podem ser usados para a filtragem. Alguns deles são passados do processo que está sendo registrado e alguns são aplicados pelo `journald` usando informações que ele coleta do sistema no momento do registro.

O sublinhado inicial indica que o campo `_PID` pertence ao segundo tipo. O **Journal** registra e classifica automaticamente o PID do processo que está sendo registrado para a filtragem posterior. É possível descobrir todos os campos de **Journal** disponíveis digitando:

```
[root@localhost 4linux]# man systemd.journal-fields
```

Por enquanto, vamos analisar mais uma opção útil relacionada com a filtragem por esses campos. A opção `-F` pode ser usada para mostrar todos os valores disponíveis para um campo de **Journal** específico.

Por exemplo, para ver quais entradas para IDs de grupo o **Journal** do `systemd` possui, digite:

```
[root@localhost 4linux]# journalctl -F _GID
```

```
32
99
102
133
81
84
100
0
124
87
```

Isso mostrará todos os valores que o **Journal** armazenou para o campo ID de grupo e pode ajudar a construir seus filtros.

Por caminho de componente

Também podemos filtrar fornecendo um caminho de pesquisa.

Se o caminho levar a um executável, o `journalctl` exibirá todas as entradas relacionadas ao executável em questão. Por exemplo, para encontrar essas entradas que estão relacionadas ao executável `bash`, digite:

```
[root@localhost 4linux]# journalctl /usr/bin/bash
```

Normalmente, se uma unidade estiver disponível para o executável, esse método é mais organizado e fornece informações mais completas (entradas de processos filhos associados, etc). Às vezes, no entanto, isso não é possível.

Exibir mensagens de kernel

As mensagens de kernel, que são aquelas geralmente encontradas na saída do `dmesg`, também podem ser recuperadas do **Journal**.

Para exibir apenas essas mensagens, podemos adicionar os sinalizadores `-k` ou `--dmesg` ao nosso comando:

```
[root@localhost 4linux]# journalctl -k
```

Por padrão, isso exibirá as mensagens do kernel da inicialização atual. É possível especificar uma outra inicialização usando os sinalizadores de seleção de inicialização discutidos anteriormente. Por exemplo, para obter as mensagens de cinco inicializações atrás, digite:

```
[root@localhost 4linux]# journalctl -k -b -5
```

Por prioridade

Um filtro no qual os administradores de sistema estão geralmente interessados é a prioridade de mensagem. Embora seja muitas vezes útil registrar informações em um nível bastante detalhado, ao resumir as informações disponíveis, o registro de baixa prioridade pode ficar confuso.

É possível usar o `journalctl` para exibir apenas mensagens de uma prioridade especificada ou superior usando a opção `-p`. Isso permite filtrar mensagens com níveis de prioridade inferiores. ‘ Por exemplo, para mostrar apenas entradas registradas no nível de erro ou acima, digite:

```
[root@localhost 4linux]# journalctl -p err -b
```

Isso exibirá todas as mensagens marcadas como erro, crítico, alerta ou emergência. O **Journal**

implementa os níveis de mensagem padrão do `syslog`. É possível usar o nome da prioridade ou seu valor numérico correspondente. As prioridades, ordenadas da maior para a menor, são:

```
0: emerg
1: alert
2: crit
3: err
4: warning
5: notice
6: info
7: debug
```

Os números ou nomes acima podem ser usados de maneira intercambiável com a `-p` opção. A seleção de uma prioridade exibirá mensagens marcadas no nível especificado e acima.

Modificar a exibição do Journal

Acima, demonstramos a seleção de entradas através da filtragem. No entanto, existem outras maneiras com as quais podemos modificar o resultado. Podemos ajustar a exibição do `journalctl` para atender diversas necessidades.

Truncar ou expandir o resultado

Podemos ajustar como o `journalctl` exibe os dados, dizendo-lhe para reduzir ou expandir o resultado.

Por padrão, o `journalctl` mostrará a entrada inteira no pager, permitindo que as entradas se expandam à direita da tela. Essa informação pode ser acessada pressionando a tecla de seta para a direita.

Se preferir o resultado truncado, inserindo reticências onde as informações foram removidas, use a opção `--no-full`:

```
[root@localhost 4linux]# journalctl --no-full
```

Também é possível ir na direção oposta e dizer ao `journalctl` para exibir todas as suas informações, mesmo se forem incluídos caracteres não imprimíveis. Podemos fazer isso com a flag `-a`:

```
[root@localhost 4linux]# journalctl -a
```

Resultados em formato padrão

Por padrão, o `journalctl` exibe o resultado em um `pager` para um consumo mais simples. No entanto, se você estiver planejando processar os dados com ferramentas de manipulação de texto, vai querer ser capaz de gerar resultados no formato padrão.

Faça isso com a opção `--no-pager`:

```
[root@localhost 4linux]# journalctl --no-pager
```

Isso pode ser canalizado imediatamente em um utilitário de processamento ou redirecionado para um arquivo no disco, dependendo das suas necessidades.

Formatos de saída

Se você estiver processando entradas do **Journal**, como mencionado acima, provavelmente será mais fácil analisar os dados se eles estiverem em um formato mais consumível. Felizmente, o **Journal** pode ser exibido em uma variedade de formatos conforme a necessidade. Faça isso usando a opção -o com um especificador de formato.

Por exemplo, gere um arquivo JSON a partir de entradas no **Journal** digitando:

```
journalctl -b -u crond -o json
{ "__CURSOR" : "s=13a21661cf4948289c63075db6c25c00;i=116f1;b=81
     b58db8fd9046ab9f847ddb82a2fa2d;m=19f0daa;t=50e33c33587ae;x=e307daadb4858635", "
     __REALTIME_TIMESTAMP" : "1422990364739502", "__MONOTONIC_TIMESTAMP" : "27200938", "
     _BOOT_ID" : "81b58db8fd9046ab9f847ddb82a2fa2d", "PRIORITY" : "6", "_UID" : "0", "_GID"
      : "0", "_CAP_EFFECTIVE" : "3fffffffff", "_MACHINE_ID" : "752737531
     a9d1a9c1e3cb52a4ab967ee", "_HOSTNAME" : "desktop", "SYSLOG_FACILITY" : "3", "CODE_FILE
     " : "src/core/unit.c", "CODE_LINE" : "1402", "CODE_FUNCTION" : "
     unit_status_log_starting_stopping_reloading", "SYSLOG_IDENTIFIER" : "systemd", "
     MESSAGE_ID" : "7d4958e842da4a758f6c1cdc7b36dcc5", "_TRANSPORT" : "journal", "_PID" :
     "1", "_COMM" : "systemd", "_EXE" : "/usr/lib/systemd/systemd", "_CMDLINE" : "/usr/lib/
     systemd/systemd", "_SYSTEMD_CGROUP" : "/", "UNIT" : "nginx.service", "MESSAGE" : "
     Starting A high performance web server and a reverse proxy server...", "
     _SOURCE_REALTIME_TIMESTAMP" : "1422990364737973" }

. . .
```

Isso é útil para análise com utilitários. Você poderia usar o formato `json-pretty` para ter uma melhor ideia melhor da estrutura de dados antes de passá-lo para o consumidor de JSON:

```
[root@localhost 4linux]# journalctl -b -u crond -o json-pretty
{
```

```
    "__CURSOR" : "s=13a21661cf4948289c63075db6c25c00;i=116f1;b=81
        b58db8fd9046ab9f847ddb82a2fa2d;m=19f0daa;t=50e33c33587ae;x=e307daadb4858635",
    "__REALTIME_TIMESTAMP" : "1422990364739502",
    "__MONOTONIC_TIMESTAMP" : "27200938",
    "_BOOT_ID" : "81b58db8fd9046ab9f847ddb82a2fa2d",
    "PRIORITY" : "6",
    "_UID" : "0",
    "_GID" : "0",
    "_CAP_EFFECTIVE" : "3fffffffff",
    "_MACHINE_ID" : "752737531a9d1a9c1e3cb52a4ab967ee",
    "_HOSTNAME" : "desktop",
    "SYSLOG_FACILITY" : "3",
    "CODE_FILE" : "src/core/unit.c",
    "CODE_LINE" : "1402",
    "CODE_FUNCTION" : "unit_status_log_starting_stopping_reloading",
    "SYSLOG_IDENTIFIER" : "systemd",
    "MESSAGE_ID" : "7d4958e842da4a758f6c1cdc7b36dcc5",
    "_TRANSPORT" : "journal",
    "_PID" : "1",
    "_COMM" : "systemd",
    "_EXE" : "/usr/lib/systemd/systemd",
    "_CMDLINE" : "/usr/lib/systemd/systemd",
    "_SYSTEMD_CGROUP" : "/",
    "UNIT" : "nginx.service",
    "MESSAGE" : "Starting A high performance web server and a reverse proxy server...",
    "_SOURCE_REALTIME_TIMESTAMP" : "1422990364737973"
}

. . .
```

Os formatos a seguir podem ser usados para exibição:

- **cat** - Exibe apenas o campo de mensagem em si.
- **export** - Um formato binário adequado para transferir e fazer um backup.
- **json** - JSON padrão com uma entrada por linha.
- **json-pretty** - JSON formatado para uma melhor legibilidade humana
- **json-sse** - Saída formatada em JSON agrupada para tornar um evento enviado ao servidor compatível
- **short** - O estilo de saída padrão do syslog
- **short-iso** - O formato padrão aumentado para mostrar as carimbos de data/hora da ISO 8601.
- **short-monotonic** - O formato padrão com carimbos de data/hora monotônicos.
- **short-precise** - O formato padrão com precisão de microssegundos
- **verbose** - Exibe todas os campos de **Journal** disponíveis para a entrada, incluindo aqueles que geralmente estão escondidos internamente.

Essas opções permitem exibir as entradas do **Journal** no formato que melhor atende às suas necessidades atuais.

Monitoramento de processo ativo

O comando journalctl imita a forma como muitos administradores usam tail para monitorar atividades ativas ou recentes. Essa funcionalidade está embutida no journalctl, permitindo acessar esses recursos sem precisar utilizar uma outra ferramenta.

Exibir registros recentes

Para exibir uma quantidade definida de registros, use a opção -n, que funciona exatamente como `tail -n`.

Por padrão, ele exibirá as 10 entradas mais recentes:

```
[root@localhost 4linux]# journalctl -n
```

Especifique o número de entradas que quer ver com um número após o -n:

```
[root@localhost 4linux]# journalctl -n 20
```

Acompanhar registros

Para acompanhar ativamente os registros à medida que são escritos, use o sinalizador -f. Novamente, isso funciona como o esperado, caso você tenha experiência usando o tail -f:

```
[root@localhost 4linux]# journalctl -f
```

Manutenção do Journal

Você deve estar se perguntando sobre o custo do armazenamento de todos os dados que vimos até agora. Além disso, você pode estar interessado em limpar alguns registros mais antigos e liberar o espaço.

Descobrir o uso atual em disco

É possível descobrir a quantidade de espaço que **Journal** está ocupando no disco usando o sinalizador `--disk-usage`:

```
```

```
[root@localhost 4linux]# journalctl --disk-usage
Journals take up 8.0M on disk.
```

Deletar registros antigos

Se quiser reduzir o tamanho do seu **Journal**, você pode fazer isso de duas maneiras diferentes (disponível com a versão 218 do systemd ou superior).

Usando a opção `--vacuum-size`, você pode reduzir o tamanho do **Journal** indicando um tamanho. Isso removerá entradas antigas até o espaço total que o **Journal** ocupa em disco esteja no tamanho solicitado:

```
[root@localhost 4linux]# journalctl --vacuum-size=1G
```

Outra maneira para reduzir o tamanho do **Journal** é fornecendo um tempo de corte com a opção –vacuum-time. Todas as entradas além daquele momento são excluídas. Isso permite manter as entradas que foram criadas após um tempo específico.

Por exemplo, para manter as entradas do último ano, digite:

```
[root@localhost 4linux]# journalctl --vacuum-time=1years
```

Limitar a expansão do Journal

É possível configurar seu servidor para colocar limites sobre a quantidade de espaço que **Journal** pode ocupar. Isso pode ser feito editando o arquivo `/etc/systemd/journald.conf`.

Os itens a seguir podem ser usados para limitar o crescimento do **Journal**:

- SystemMaxUse=: especifica o espaço máximo em disco que pode ser usado pelo **Journal** em armazenamento persistente.
- SystemKeepFree=: especifica a quantidade de espaço que o **Journal** deve deixar livre no armazenamento persistente ao adicionar entradas.
- SystemMaxFileSize=: controla o tamanho máximo até o qual os arquivos de **Journal** individuais podem crescer em armazenamento persistente antes de serem girados.
- RuntimeMaxUse=: especifica o espaço máximo em disco que pode ser usado como armazenamento volátil (dentro do sistema de arquivos /run).
- RuntimeKeepFree=: especifica a quantidade de espaço a ser reservada para outros usos

ao escrever dados no armazenamento volátil (dentro do sistema de arquivos /run).
- RuntimeMaxFileSize=: especifica a quantidade de espaço que um arquivo de **Journal** individual pode ocupar em armazenamento volátil (dentro do sistema de arquivos /run) antes de ser girado.

Ao definir esses valores, você pode controlar como o journald consome e preserva o espaço no seu servidor. Tenha em mente que o `SystemMaxFileSize` e o `RuntimeMaxFileSize` vão mirar em arquivos arquivados para alcançar os limites declarados. Isso é importante de lembrar ao interpretar contagens de arquivos após uma operação de limpeza. # Preservar os Journals do sistema

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - Inicialize sistemas em diferentes alvos manualmente
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - Identifique processos intensivos de CPU/memória e elimine processos
  - Ajuste a programação de processos
  - Gerencie perfis de ajuste
  - Localize e interprete os arquivos de registro do sistema e diários
  - **Preserve diários do sistema**
  - Inicie, pare e verifique o status dos serviços de rede
  - Transfira arquivos com segurança entre sistemas

Systemd

Nesta aula, veremos como você pode persistir os logs de Journal do Systemd em seu servidor. Preservar o Journal do sistema pode ser útil para solucionar problemas de seus serviços quando as coisas estão sempre quebrando.

Noções básicas de Journals do sistema

O Systemd, por padrão, armazena os Journals do sistema no diretório `/run/log/journal`. Tudo no diretório `/run` é limpo e o conteúdo é recriado na reinicialização. Isso significa que os diários são apagados quando o sistema é reinicializado. Podemos ajustar as definições de configuração do serviço `systemd-journald` no arquivo `/etc/systemd/journald.conf` para fazer os diários persistirem durante a reinicialização.

Abra o arquivo para visualizar seu conteúdo:

```
[root@localhost 4linux]# cat  /etc/systemd/journald.conf
```

Configurando Journal do Sistema Persistente

Para configurar o serviço systemd-journald e preservar os Journal do sistema persistentemente durante a reinicialização, você precisa definir `Storage` para `persistent`:

```
[Journal]
Storage=persistent
```

Outros valores que podem ser definidos para o parâmetro de armazenamento são:

- **persistent** - Armazena Journals no diretório `/var/log/journal` que persiste nas reinicializações. O diretório é criado se não existir.
- **volatile** - Armazena Journals no diretório volátil `/run/log/journal`. Isso não persiste nas reinicializações do sistema.
- **auto** - O `rsyslog` determinará se deve usar armazenamento persistente ou volátil.

Assim que as alterações forem confirmadas, reinicie o serviço systemd-journald para que as alterações de configuração tenham efeito.

```
[root@localhost 4linux]# systemctl restart systemd-journald
```

O diretório `/var/log/journal` deve ser criado. Os subdiretórios em `/var/log/journal` têm caracteres hexadecimais em seus nomes longos e contêm `*.journalfiles`. Esses são os arquivos binários que armazenam as entradas de diário estruturadas e indexadas.

Ajustando o armazenamento para diários

Você pode definir o tamanho máximo do diário persistente removendo o comentário e alterando o seguinte:

```
SystemMaxUse=500M
```

O limite de tamanho padrão é definido como um valor de 10% do tamanho do sistema de arquivos subjacente, mas limitado a 4 GiB.# Inicie, pare e verifique o status dos serviços de rede ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - Inicialize sistemas em diferentes alvos manualmente
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - Identifique processos intensivos de CPU/memória e elimine processos
  - Ajuste a programação de processos
  - Gerencie perfis de ajuste
  - Localize e interprete os arquivos de registro do sistema e diários
  - Preserve diários do sistema
  - **Inicie, pare e verifique o status dos serviços de rede**
  - Transfira arquivos com segurança entre sistemas

Systemd

O systemd é um sistema de inicialização (init system) composto por um conjunto de programas executado em segundo plano (ou seja, um daemon). Atualmente, a maioria das distribuições Linux utilizam do systemd para execução do boot.

Desde a época em que foi lançado, o systemd trouxe uma rica quantidade de recursos e funcionalidades que, em comparação aos tradicionais sysvinit e Upstart, oferece muito mais possibilidades. Inclusive, ele não é considerado um sistema de init, somente.

Outra vantagem do systemd é a sua arquitetura e modo de funcionamento. Nela são usadas unidades de socket, que são arquivos de configuração que codificam informações relacionadas à comunicação entre processos (Inter Process Communication – IPC), a soquetes de rede ou a arquivos FIFO.

Qual a importância disso tudo? Tal capacidade permite que todos os daemons requisitados no boot sejam carregados simultaneamente, bem como possibilita a transmissão coordenada entre dois sockets, resultando a rápida inicialização do sistema operacional.

Para efeitos de comparação, o sysvinit, utilizado em distribuições mais antigas do Linux, carrega todos os serviços (um por vez, automática e ordenadamente), utilizando shell scripts durante o boot. Em razão disso, ele é um sistema lento e propício a problemas.

De que "problemas" estamos falando? Bom, vamos supor que determinado serviço não foi carregado pelo sysvinit. Diante da situação, o usuário precisa iniciá-lo manualmente, pois o

sysvinit não pode ser acionado depois de concluído o boot.

A princípio, pode até não parecer muito, mas, se colocarmos à mesa todos os dispositivos que conectamos no computador por meio de portas USB, LAN, HDMI, adaptador Bluetooth, entre outros, percebemos que as dores de cabeça seriam bastante recorrentes.

Você se lembra de quando mencionamos o systemd como evolução (e revolução, ao mesmo tempo) desse tipo de sistema? Pois então, essa adaptabilidade às tecnologias atuais justifica o que eu disse.

Arquitetura do systemd

Basicamente, a estrutura do systemd parte da organização de suas tarefas em unidades (units). Abaixo, algumas classes de units que constituem o systemd:

- **service** - São os serviços presentes no sistema operacional acessíveis ao usuário;
- **timer** - Temporizadores usados para determinar ações para um serviço, usando como base o tempo (não confundir com o cron);
- **mount** - Arquivo de configuração que codifica informações sobre um diretório controlado e supervisionado pelo systemd;
- **target** - Grupos de unidades que reúnem todas as units necessárias para iniciar um determinado serviço;
- **snapshot** - Mecanismo usado para criar snapshots dinâmicos do estado atual do systemd manager, útil para retomar o estado após problemas com indisponibilidade, por exemplo;
- **path** - Unidades especialmente utilizadas para monitorar arquivos e diretórios para eventos e, também, executar serviços;
- **socket** - Arquivo de configuração que armazena informações acerca de um IPC ou soquete de rede ou arquivo FIFO; e
- **swap** - Guarda informações relativas a dispositivos usados para swapping, bem como serviços que utilizam de memória Swap.

Cada serviço é alocado pelo systemd em um grupo de controle dedicado (control group, ou cgroup). No cgroup são organizadas informações voltadas aos processos que fazem parte do grupo, como limite, supervisão e contabilização de recursos computacionais que eles consomem.

O controle desses grupos é feito a partir de utilitários que acompanham o systemd, tais como: journalctl, cgls, cgtop e systemctl — ainda falaramos sobre este último, aqui. Mesmo com uma apresentação bem superficial dos seus componentes, é possível ter noção de quão robusto é o systemd.

Iniciando e interrompendo serviços

Para iniciar um serviço `systemd`, executando instruções no arquivo de unidade do serviço, use o comando `start`. Se você estiver executando como um usuário não root, terá que usar o `sudo`, pois isso afetará o estado do sistema operacional:

```
[root@localhost 4linux]# systemctl start application.service
```

O `systemd` sabe como procurar por arquivos `.service` para comandos de gerenciamento de serviço, então o comando poderia ser facilmente digitado assim:

```
[root@localhost 4linux]# systemctl start application
```

Embora você possa usar o formato acima para administração geral, para maior clareza, usaremos o sufixo `.service` e para o restante dos comandos para ser explícito sobre o destino no qual estamos operando.

Para interromper um serviço em execução, você pode usar o comando `stop`:

```
[root@localhost 4linux]# systemctl stop application.service
```

Reiniciando e recarregando

Para reiniciar um serviço em execução, você pode usar o comando `restart`:

```
[root@localhost 4linux]# systemctl restart application.service
```

Se o aplicativo em questão for capaz de recarregar seus arquivos de configuração (sem reiniciar), você pode emitir o comando `reload` para iniciar esse processo:

```
[root@localhost 4linux]# systemctl reload application.service
```

Se não tiver certeza se o serviço tem a funcionalidade de recarregar sua configuração, você

pode emitir o comando `reload-or-restart`. Isso recarregará a configuração local, se disponível. Caso contrário, ele reiniciará o serviço para que a nova configuração seja selecionada:

```
[root@localhost 4linux]# systemctl reload-or-restart application.service
```

Ativando e desativando serviços

Os comandos acima são úteis para iniciar ou interromper comandos durante a sessão atual. Para dizer ao `systemd` para iniciar os serviços automaticamente na inicialização, você deve habilitá-los. Para iniciar um serviço na inicialização, use o comando `enable`:

```
[root@localhost 4linux]# systemctl enable application.service
```

Isso criará um link simbólico da cópia do sistema do arquivo de serviço geralmente em `/lib/systemd/system` ou `/etc/systemd/system` para o local no disco onde o `systemd` procura por arquivos de inicialização automática geralmente `/etc/systemd/system/some_target.target.wants`.

Para impedir que o serviço seja iniciado automaticamente, você pode digitar:

```
[root@localhost 4linux]# systemctl disable application.service
```

Isso removerá o link simbólico que indicava que o serviço deve ser iniciado automaticamente. Lembre-se de que habilitar um serviço não o inicia na sessão atual. Se desejar iniciar o serviço e habilitá-lo na inicialização, você terá que emitir os comandos `start` e `enable`.

Verificando o status dos serviços

Para verificar o status de um serviço em seu sistema, você pode usar o comando `status`:

```
[root@localhost 4linux]# systemctl status application.service
```

Isso fornecerá o estado do serviço, a hierarquia do cgroup e as primeiras linhas de log.

Isso oferece uma boa visão geral do status atual do aplicativo, notificando-o sobre quaisquer problemas e ações que possam ser necessárias.

Também existem métodos para verificar estados específicos. Por exemplo, para verificar se uma unidade está atualmente ativa (em execução), você pode usar o comando `is-active`:

```
[root@localhost 4linux]# systemctl is-active application.service
```

Isso retornará o estado atual da unidade, que geralmente é `active` ou `inactive`. O código de saída será `0` se estiver ativo, tornando o resultado mais simples de analisar em scripts de shell.

Para ver se a unidade está habilitada, você pode usar o comando `is-enabled`:

```
[root@localhost 4linux]# systemctl is-enabled application.service
```

Isso resultará em se o serviço é `enabled` ou `disabled` e configurará novamente o código de saída para `0` ou `1`, dependendo da resposta à pergunta do comando.

Uma terceira verificação é se a unidade está em um estado de falha. Isso indica que houve um problema ao iniciar a unidade em questão:

```
[root@localhost 4linux]# systemctl is-failed application.service
```

Isso retornará `active` se estiver funcionando corretamente ou `failed` se ocorrer um erro. Se a unidade foi parada intencionalmente, ela pode retornar `unknown` ou `inactive`. Um status de saída `0` indica que ocorreu uma falha e um status de saída `1` indica qualquer outro status.

11

Transferir de forma segura arquivos entre sistemas

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Operar sistemas em execução**
  - Inicialize, reinicie e desligue um sistema normalmente
  - Inicialize sistemas em diferentes alvos manualmente
  - Interrompa o processo de inicialização para obter acesso a um sistema
  - Identifique processos intensivos de CPU/memória e elimine processos
  - Ajuste a programação de processos
  - Gerencie perfis de ajuste
  - Localize e interprete os arquivos de registro do sistema e diários
  - Preserve diários do sistema
  - Inicie, pare e verifique o status dos serviços de rede
  - **Transfira arquivos com segurança entre sistemas**

Introdução

SCP (cópia segura) é um utilitário de linha de comando que permite copiar arquivos e diretórios com segurança entre dois locais.

Com o scp, você pode copiar um arquivo ou diretório:

- Do seu sistema local para um sistema remoto.
- De um sistema remoto para o seu sistema local.
- Entre dois sistemas remotos de seu sistema local.

Ao transferir dados com scp, os arquivos e a senha são criptografados para que qualquer pessoa que esteja espionando o tráfego não receba nada confidencial.

Sintaxe Comando SCP

Antes de entrar em como usar o comando scp, vamos começar revisando a sintaxe básica. A sintaxe scp do comando assume o seguinte formato:

```
scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2
```

- **OPTION** - Opções de scp como cifra, configuração ssh, porta ssh, limite, cópia recursiva ... etc.
- **[user@]SRC_HOST:]file1** - Arquivo fonte.
- **[user@]DEST_HOST:]file2** - Arquivo de destino.

Os arquivos locais devem ser especificados usando um caminho absoluto ou relativo, enquanto os nomes dos arquivos remotos devem incluir uma especificação de usuário e host. O scp fornece várias opções que controlam todos os aspectos de seu comportamento. As opções mais utilizadas são:

- **-P** - Especifica a porta SSH do host remoto.
- **-p** - Preserva a modificação de arquivos e tempos de acesso.
- **-q** - Use esta opção se desejar suprimir o medidor de progresso e as mensagens de não erro.
- **-C** - Esta opção força o scp a compactação dos dados à medida que são enviados para a máquina de destino.
- **-r** - Esta opção informa para o scp copiar diretórios recursivamente.

Copie arquivos e diretórios entre dois sistemas com scp

Copie um arquivo local para um sistema remoto com o comando `scp`. Para copiar um arquivo de um sistema local para um remoto, execute o seguinte comando:

```
[root@localhost 4linux]# scp arquivo.txt usuario_remoto@10.10.0.2:/diretorio/remoto
```

Onde `arquivo.txt` é o nome do arquivo que queremos copiar, `usuario_remoto` é o usuário no servidor remoto, `10.10.0.2` é o endereço IP do servidor.

O `/diretorio/remoto` é o caminho para o diretório que você deseja copiar o arquivo. Se você não especificar um diretório remoto, o arquivo será copiado para o diretório inicial do usuário remoto.

Será solicitado que você insira a senha do usuário e o processo de transferência será iniciado.

```
usuario_remoto@10.10.0.2's password:
arquivo.txt                                   100%    0     0.0KB/s   00:00
```

A omissão do nome do arquivo do local de destino copia o arquivo com o nome original. Se você deseja salvar o arquivo com um nome diferente, é necessário especificar o novo nome do arquivo:

```
scp arquivo.txt usuario_remoto@10.10.0.2:/diretorio/remoto/newfilename.txt
```

Se o `SSH` no host remoto estiver escutando em uma porta diferente da `22` padrão, você pode especificar a porta usando o argumento `-P`:

```
[root@localhost 4linux]# scp -P 2322 arquivo.txt usuario_remoto@10.10.0.2:/diretorio/remoto
```

O comando para copiar um diretório é muito parecido com o da cópia de arquivos. A única diferença é que você precisa usar a flag `-r` para recursivo.

Para copiar um diretório de um sistema local para remoto, use a opção `-r`:

```
[root@localhost 4linux]# scp -r /diretorio/local usuario_remoto@10.10.0.2:/diretorio/remoto
```

Copie um arquivo remoto para um sistema local usando o comando

Para copiar um arquivo de um sistema remoto para um local, use o local remoto como origem e o local como destino. Por exemplo, para copiar um arquivo nomeado `arquivo.txt` de um servidor remoto com IP, `10.10.0.2` execute o seguinte comando:

```
[root@localhost 4linux]# scp usuario_remoto@10.10.0.2:/remoto/arquivo.txt /diretorio/local
```

Se você não configurou um login `SSH` sem senha para a máquina remota, será solicitado que você digite a senha do usuário.

Copie um arquivo entre dois sistemas remotos usando o comando

Ao contrário do comando `rsync`, ao usar, `scp` você não precisa fazer login em um dos servidores para transferir arquivos de uma para outra máquina remota.

O comando a seguir copiará o arquivo `/arquivos/arquivo.txt` do host remoto `host1.com` para o diretório `/arquivos` no host remoto `host2.com`.

```
[root@localhost 4linux]# scp user1@host1.com:/arquivos/arquivo.txt user2@host2.com:/
     arquivos
```

Você será solicitado a inserir as senhas de ambas as contas remotas. Os dados serão transferidos diretamente de um host remoto para outro.

Para rotear o tráfego por meio da máquina na qual o comando é emitido, use a opção `-3`:

```
[root@localhost 4linux]# scp -3 user1@host1.com:/arquivos/arquivo.txt user2@host2.com:/
     arquivos
```

SFTP

(SSH File Transfer Protocol) é um protocolo de arquivo seguro usado para acessar, gerenciar e transferir arquivos por meio de um transporte SSH criptografado.

Quando comparado com o protocolo FTP tradicional, o SFTP oferece todas as funcionalidades do FTP, mas é mais seguro e fácil de configurar.

Ao contrário do SCP, que suporta apenas transferências de arquivos, o SFTP permite que você execute uma variedade de operações em arquivos remotos e retome as transferências de arquivos.

Estabelecendo uma conexão SFTP

SFTP funciona em um modelo cliente-servidor. É um subsistema de SSH e suporta todos os mecanismos de autenticação SSH.

Para abrir uma conexão SFTP com um sistema remoto, use o comando `sftp` seguido pelo nome de usuário do servidor remoto e o endereço IP ou nome de domínio:

```
[root@localhost 4linux]# sftp usuario_remoto@server_ip_or_hostname
```

Se você estiver se conectando ao host usando autenticação de senha, será solicitado que você insira a senha do usuário.

Depois de conectado, será apresentado um prompt `sftp` e você poderá começar a interagir com o servidor remoto:

```
Connected to usuario_remoto@server_ip_or_hostname.
sftp>
```

Se o servidor SSH remoto não estiver escutando na porta `22` padrão , use a opção `-P` para especificar a porta SFTP:

```
[root@localhost 4linux]# sftp -P custom_port usuario_remoto@server_ip_or_hostname
```

Comandos SFTP

A maioria dos comandos `SFTP` são semelhantes ou idênticos aos comandos do shell do Linux. Para obter uma lista de todos os comandos `SFTP` disponíveis, digite `help` ou `?`.

```
sftp> help
```

Isso resultará em uma longa lista de todos os comandos disponíveis, incluindo uma breve descrição de cada comando:

```
Available commands:
bye                                Quit sftp
cd path                            Change remote directory to 'path'
...
...
version                            Show SFTP version
!command                           Execute 'command' in local shell
!                                  Escape to local shell
?                                  Synonym for help
```

Navegando com SFTP

Quando você está conectado ao servidor remoto, seu diretório de trabalho atual é o diretório inicial do usuário remoto. Você pode verificar isso digitando:

```
sftp> pwd
```

```
Remote working directory: /home/usuario_remoto
```

Para listar os arquivos e diretórios, use o comando `ls`:

```
sftp> ls
```

Para navegar para outro diretório, use o comando `cd`. Por exemplo, para mudar para o diretório `/tmp`, você digitaria:

```
sftp> cd /tmp
```

Os comandos acima são usados para navegar e trabalhar no local remoto. O shell SFTP também fornece comandos para navegação local, informações e gerenciamento de arquivos. Os comandos locais são prefixados com a letra `l`.

Por exemplo, para imprimir o diretório de trabalho local, você digitaria:

```
sftp> cd lpwd
Local working directory: /home/local_username
```

Transferindo arquivos com SFTP

SFTP permite transferir arquivos entre duas máquinas com segurança. Se você estiver trabalhando em uma máquina desktop, você pode usar um cliente SFTP GUI, como WinSCP ou FileZilla, para se conectar ao servidor remoto e baixar ou enviar arquivos.

O comando `sftp` é útil quando você trabalha em um servidor sem GUI e deseja transferir arquivos ou realizar outras operações nos arquivos remotos.

Baixando arquivos com o comando SFTP

Para baixar um único arquivo do servidor remoto, use o getcomando:

```
sftp> get filename.zip
```

A saída deve ser semelhante a esta:

```
Fetching /home/usuario_remoto/filename.zip to filename.zip
/home/usuario_remoto/filename.zip                          100%   24MB   1.8MB/s   00:13
```

Ao baixar arquivos com `sftp`, os arquivos são baixados para o diretório no qual você digitou o comando `sftp`.

Se você deseja salvar o arquivo baixado com um nome diferente, especifique o novo nome

como o segundo argumento:

```
sftp> get filename.zip local_filename.zip
```

Para baixar um diretório do sistema remoto, use a opção recursiva `-r`:

```
sftp> get -r remote_directory
```

Se a transferência de um arquivo falhar ou for interrompida, você pode retomá-la usando o comando `reget`.

A sintaxe de regeté igual à sintaxe de get:

```
sftp> reget filename.zip
```

Upload de arquivos com o comando SFTP

Para fazer upload de um arquivo da máquina local para o servidor SFTP remoto, use o comando `put`:

```
sftp> put filename.zip
```

A saída deve ser semelhante a esta:

```
Uploading filename.zip to /home/usuario_remoto/filename.zip
filename.zip                          100%   12MB   1.7MB/s   00:06
```

Se o arquivo que você deseja enviar não estiver localizado no diretório de trabalho atual, use o caminho absoluto para o arquivo.

Ao trabalhar com `put` você pode usar as mesmas opções disponíveis com o comando `get`.

Para fazer upload de um diretório local, você digitaria:

```
sftp> put -r locale_directory
```

Para retomar um upload interrompido:

```
sftp> reput filename.zip
```

Manipulações de arquivo com SFTP

Normalmente, para realizar tarefas em um servidor remoto, você se conectaria a ele via SSH e faria seu trabalho usando o terminal shell. No entanto, em algumas situações, o usuário pode ter apenas acesso SFTP ao servidor remoto.

O SFTP permite que você execute alguns comandos básicos de manipulação de arquivos. Abaixo estão alguns exemplos de como usar o shell SFTP:

- Obtenha informações sobre o uso de disco do sistema remoto:

```
sftp> df
        Size         Used        Avail       (root)    %Capacity
    20616252      1548776     18002580     19067476           7%cópia de
```

- Crie um novo diretório no servidor remoto:

```
sftp> mkdir directory_name
```

- Renomeie um arquivo no servidor remoto:

```
sftp> rename file_name new_file_name
```

- Exclua um arquivo no servidor remoto:

```
sftp> rm file_name
```

- Exclua um diretório no servidor remoto:

```
sftp> rmdir directory_name
```

- Altere as permissões de um arquivo no sistema remoto:

```
sftp> chmod 644 file_name
```

- Altere o proprietário de um arquivo no sistema remoto:

```
sftp> chown user_id file_name
```

Você deve fornecer o ID do usuário para os comandos `chown` e `chgrp`.

- Altere o proprietário do grupo de um arquivo remoto com:

```
sftp> chgrp group_id file_name
```

Quando terminar seu trabalho, feche a conexão digitando `bye` ou `quit`.# Listar, criar e excluir partições em discos MBR e GPT ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias.

- **Configure o armazenamento local**
  - **Listar, criar e excluir partições em discos MBR e GPT**
  - Criar e remover volumes físicos
  - Atribuir volumes físicos a grupos de volume
  - Criar e excluir volumes lógicos
  - Configurar os sistemas para montagem de sistemas de arquivos na inicialização por ID universalmente exclusivo (UUID) ou rótulo
  - Adicionar novas partições e volumes lógicos e mudar para um sistema de forma não destrutiva

Introdução

Um disco pode ser usado como uma entidade simples ou dividido em uma ou mais partições.

Os discos são geralmente chamados de `/dev/sda`, `/dev/sdb` etc.

Já as partições obtêm seus nomes do próprio nome do disco e adicionam um número começando em `1` (`/dev/sda1`, `/dev/sda2`).

Uma tabela de partição é uma estrutura especial que contém a organização de partições.

Discos não recentes usam setores de 512 bytes e a tabela de partição `MBR` (que significa Master Boot Record). Esta organização permite apenas 4 partições primárias. Se você quiser mais do que isso, precisará criar uma partição estendida (usando um dos 4 slots principais) e, em seguida, criar partições lógicas dentro dela. Mais irritante, em discos com capacidade superior a 2 TB, o espaço acima desse limite não está disponível.

Para contornar todas estas limitações, discos recentes usam setores de 4096 bytes e o `GPT` tabela de partição (que significa GUID - Globally Unique IDentifier - Partition Table).

Historicamente, existem dois comandos para manipular discos e partições: `fdisk` e `parted` . Como o comando `fdisk` não lida com tabelas de partição `GPT`, não é aconselhável usá-lo mais. Recentemente, uma nova ferramenta chamada `gdisk` foi criada para lidar com tabelas de partição `GPT`, oferecendo uma alternativa ao comando `parted`.

```
[root@localhost 4linux]# parted
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Partições de disco

Primeiro, usaremos a configuração clássica do `MBR` (Master Boot Record), depois faremos o mesmo na configuração do `GPT` (GUID Partitioning Table).

Gerenciando partições com `fdisk`

O particionamento é a primeira etapa para expandir o espaço em disco que pode ser usado pelo sistema para armazenar dados. Abordaremos o particionamento `MBR` e `GPT`, criando, listando e, finalmente, excluindo partições.

Listar partições

Para listar partições, podemos usar `fdisk`. Faremos isso para ver nossa configuração no início.

```
[root@localhost 4linux]# fdisk -l
```

Provavelmente, na saída, podemos ver que temos o disco principal `/dev/sda` com duas partições, `/dev/sda1` e `/dev/sda2`. Podemos também ver nosso novo vazio `/dev/sdb` sem partições ainda, bem como os volumes lógicos que o sistema contém.

Crie uma partição

Para criar uma nova partição no disco vazio, vamos fornecê-la como argumento para `fdisk`:

```
[root@localhost 4linux]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n

Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p

Partition number (1-4, default 1):
First sector (2048-4194303, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.
```

O primeiro e o último setor determinarão o tamanho real da partição. Em nosso exemplo, estamos criando uma única partição que cobrirá o disco e os valores padrão são a primeira partição, o primeiro setor disponível para iniciar e o último setor para terminar, que é exatamente o que precisamos. Não nos limitamos a contar em setores quando definimos o fim da partição. Como o utilitário sugere, podemos especificar um tamanho exato. Por exemplo, se quisermos uma partição de 1 GB de tamanho (em vez de 2 GB), no último setor podemos fornecer:

```
Last sector, +sectors or +size{K,M,G,T,P} (2048-4194303, default 4194303):
```

A partição agora está completa, mas como o `fdisk` indica no início, as alterações ficam na memória apenas até que as gravemos no disco. Isso é proposital e o aviso existe por um bom motivo: ao gravar as alterações no disco, destruímos tudo o que residia na faixa de setor que cobrimos com nossa nova partição. Temos certeza de que não haverá perda de dados, por isso gravamos as alterações no disco:

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Para ter certeza de que o sistema operacional sabe das mudanças, rodamos o comando `partprobe`. Podemos usar o recurso `fdisk -l` para ser mais específico, adicionando o nome do dispositivo no qual estamos interessados.

```
[root@localhost 4linux]# fdisk -l /dev/sdb

Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x29ccc11b

Device     Boot Start     End Sectors Size Id Type
/dev/sdb1        2048 4194303 4192256   2G 83 Linux
```

Apagar uma Partição

A exclusão da partição é basicamente o mesmo processo ao contrário. O utilitário é construído de forma lógica: especificamos o dispositivo com o qual gostaríamos de trabalhar, e quando selecionamos a exclusão da partição com o comando `d`, ele excluirá nossa única partição sem qualquer dúvida, porque há apenas uma no disco.

```
[root@localhost 4linux]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): d
Selected partition 1
Partition 1 has been deleted.
```

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
[root@localhost 4linux]# partprobe
```

```
[root@localhost 4linux]# fdisk -l /dev/sdb
Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x29ccc11b
```

Embora seja bastante conveniente, observe que essa ferramenta torna realmente fácil limpar os dados do disco com um único pressionamento de tecla. É por isso que todos os avisos estão em vigor: você tem que saber o que está fazendo. As proteções ainda estão em vigor, nada muda no disco até que escrevamos.

Gerenciando partições com GPT

Crie uma partição

Para criar um layout de partição baseado em `GPT`, usaremos o utilitário `gdisk` (GPT fdisk).

```
[root@localhost 4linux]# gdisk /dev/sdb

GPT fdisk (gdisk) version 1.0.3

Partition table scan:
  MBR: MBR only
  BSD: not present
  APM: not present
  GPT: not present


****************************************************************
Found invalid GPT and valid MBR; converting MBR to GPT format
in memory. THIS OPERATION IS POTENTIALLY DESTRUCTIVE! Exit by
typing 'q' if you don't want to convert your MBR partitions
to GPT format!
****************************************************************


Command (? for help): n
Partition number (1-128, default 1):
```

```
First sector (34-4194270, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-4194270, default = 4194270) or {+-}size{KMGTP}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/sdb.
The operation has completed successfully.
```

Listar partições GPT

Listar partições `GPT` requer a mesma mudança para `gdisk`:

```
[root@localhost 4linux]# gdisk -l /dev/sdb

GPT fdisk (gdisk) version 1.0.3

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
Disk /dev/sdb: 4194304 sectors, 2.0 GiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 3AA3331F-8056-4C3E-82F3-A67254343A05
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 4194270
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048         4194270   2.0 GiB     8300  Linux filesystem
```

Deletar partição GPT

A exclusão da partição `GPT` que criamos é feita de forma semelhante ao caso MBR, com a verificação de integridade adicional adicionada:

```
[root@localhost 4linux]# gdisk /dev/sdb

GPT fdisk (gdisk) version 1.0.3
```

```
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): d
Using 1

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/sdb.
The operation has completed successfully.
```

12

Crie e remova volumes físicos

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias.

- **Configure o armazenamento local**
  - Listar, criar e excluir partições em discos MBR e GPT
  - **Criar e remover volumes físicos**
  - Atribuir volumes físicos a grupos de volume
  - Criar e excluir volumes lógicos
  - Configurar os sistemas para montagem de sistemas de arquivos na inicialização por ID universalmente exclusivo (UUID) ou rótulo
  - Adicionar novas partições e volumes lógicos e mudar para um sistema de forma não destrutiva

Introdução

LVM, ou *Logical Volume Management*, é uma tecnologia de gerenciamento de dispositivo de armazenamento que dá aos usuários o poder de agrupar e abstrair o layout físico dos dispositivos de armazenamento de componentes para uma administração mais fácil e flexível. Utilizando a estrutura do kernel Linux do mapeador de dispositivos, a iteração atual, LVM2, pode ser usada para reunir dispositivos de armazenamento existentes em grupos e alocar unidades lógicas do

espaço combinado conforme necessário.

As principais vantagens do LVM são abstração, flexibilidade e controle aumentados. Os volumes lógicos podem ter nomes significativos como **bancos de dados** ou **backup raiz**. Os volumes podem ser redimensionados dinamicamente conforme os requisitos de espaço mudam e migrados entre dispositivos físicos dentro do pool em um sistema em execução ou exportados facilmente. O LVM também oferece recursos avançados como instantâneo, distribuição e espelhamento.

Nessa aula, discutiremos brevemente como o LVM funciona e, em seguida, demonstraremos os comandos básicos necessários para começar a funcionar rapidamente.

Arquitetura e Terminologia LVM

Antes de mergulharmos nos comandos administrativos reais do LVM, é importante ter um entendimento básico de como o LVM organiza os dispositivos de armazenamento e um pouco da terminologia que ele emprega.

Estruturas de gerenciamento de armazenamento LVM

O LVM funciona colocando abstrações em camadas sobre os dispositivos de armazenamento físico. As camadas básicas que o LVM usa, começando com as mais primitivas, são:

- **Volumes físicos**:
  - Prefixo do utilitário LVM:pv...
  - Descrição: dispositivos de bloco físico ou outros dispositivos semelhantes a discos (por exemplo, outros dispositivos criados pelo mapeador de dispositivos, como matrizes RAID) são usados pelo LVM como o material de construção bruto para níveis mais altos de abstração. Os volumes físicos são dispositivos de armazenamento regulares. O LVM grava um cabeçalho no dispositivo para alocá-lo para gerenciamento.
- **Grupos de volume**:
  - Prefixo do utilitário LVM:vg...
  - Descrição: o LVM combina volumes físicos em pools de armazenamento conhecidos como grupos de volumes. Os grupos de volumes abstraem as características dos dispositivos subjacentes e funcionam como um dispositivo lógico unificado com capacidade de armazenamento combinada dos volumes físicos componentes.
- **Volumes lógicos**:
  - Prefixo do utilitário LVM: lv... (utilitários LVM genéricos podem começar com lvm...)
  - Descrição: um grupo de volume pode ser dividido em qualquer número de vol-

umes lógicos. Os volumes lógicos são funcionalmente equivalentes às partições em um disco físico, mas com muito mais flexibilidade. Os volumes lógicos são o componente principal com o qual os usuários e aplicativos vão interagir.

Em resumo, o LVM pode ser usado para combinar volumes físicos em grupos de volume para unificar o espaço de armazenamento disponível em um sistema. Posteriormente, os administradores podem segmentar o grupo de volume em volumes lógicos arbitrários, que atuam como partições flexíveis.

Extensões?

Cada volume dentro de um grupo de volume é segmentado em pequenos pedaços de tamanho fixo chamados **extensões**. O tamanho das extensões é determinado pelo grupo de volumes (todos os volumes dentro do grupo estão em conformidade com o mesmo tamanho de extensão).

As extensões em um volume físico são chamadas de extensões físicas, enquanto as extensões de um volume lógico são chamadas de extensões lógicas. Um volume lógico é simplesmente um mapeamento que o LVM mantém entre as extensões lógicas e físicas. Por causa desse relacionamento, o tamanho da extensão representa a menor quantidade de espaço que pode ser alocada pelo LVM.

As extensões estão por trás de grande parte da flexibilidade e do poder do LVM. As extensões lógicas que são apresentadas como um dispositivo unificado pelo LVM não precisam ser mapeadas para extensões físicas contínuas. O LVM pode copiar e reorganizar as extensões físicas que compõem um volume lógico sem qualquer interrupção para os usuários. Volumes lógicos também podem ser facilmente expandidos ou reduzidos simplesmente adicionando extensões ou removendo extensões do volume.

Marque os dispositivos físicos como volumes físicos

Nossa primeira etapa é verificar o sistema em busca de dispositivos de bloco que o LVM possa ver e gerenciar. Você pode fazer isso digitando:

```
[root@localhost joatham]# lvmdiskscan
  /dev/sda1 [       1,00 GiB]
  /dev/sda2 [     <10,00 GiB] LVM physical volume
  /dev/sdb  [       2,00 GiB]
  /dev/sdc  [       2,00 GiB]
  2 disks
  1 partition
  0 LVM physical volume whole disks
  1 LVM physical volume
```

A saída exibirá todos os dispositivos de bloco disponíveis com os quais o LVM pode interagir:

> Certifique-se de verificar novamente se os dispositivos que você pretende usar com o LVM não possuem dados importantes já gravados neles. Usar esses dispositivos dentro do LVM sobrescreverá o conteúdo atual. Se você já possui dados importantes em seu servidor, faça backups antes de prosseguir.

Para esta aula, vamos adicionar dois discos `/dev/sdb` e `/dev/sdc` em nosso sistema, enquanto `/dev/sda` mantém o sistema operacional, que não tocaremos durante as etapas a seguir.

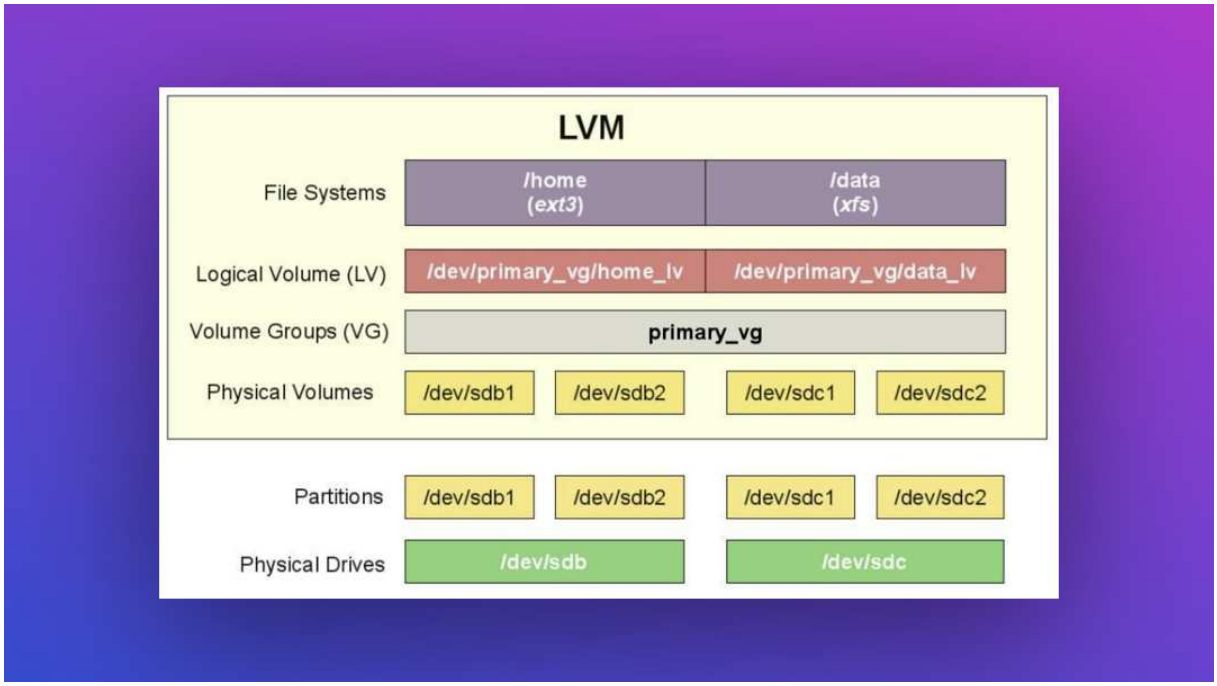


Fig. 12.1: LVM

Como criar volume físico

Para criar um volume físico, usaremos o comando `pvcreate`. Vamos criar um volume físico de `2GB` em ambos `sdb`, e `sdc`.

```
[root@localhost 4linux]# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created.

[root@localhost 4linux]# pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created.
```

Podemos verificar se o fizemos corretamente listando nossos volumes físicos. Usaremos o `pvdisplay`.

```
[root@localhost joatham]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda2
  VG Name               rhel
  PV Size               <10,00 GiB / not usable 3,00 MiB
  Allocatable           yes (but full)
  PE Size               4,00 MiB
  Total PE              2559
  Free PE               0
  Allocated PE          2559
  PV UUID               qYRZNa-m2GI-50n7-evle-KXD3-LjNs-STqWT0

  "/dev/sdb" is a new physical volume of "2,00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb
  VG Name
  PV Size               2,00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               v1fHkz-fmcs-LH9z-cV1j-WCMm-Bd8G-4fv0vs

  "/dev/sdc" is a new physical volume of "2,00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdc
  VG Name
  PV Size               2,00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               FKivAc-STxF-lbb6-bo5i-eTlf-l3yf-rmFTsW
```

Remova o volume físico

Para deletar um volume físico, temos o `pvremove` comando. Não remova um volume físico com dados gravados nele que sejam necessários.

```
[root@localhost 4linux]# pvremove /dev/sdc
  Labels on physical volume "/dev/sdc" successfully wiped.
```

Você pode verificar rapidamente se o `LVM` registrou os volumes físicos digitando:

“‘shell [root@localhost 4linux]# pvs

Output PV VG Fmt Attr PSize PFree /dev/sda2 rhel lvm2 a– <10,00g 0 /dev/sdb lvm2 — 2,00g 2,00g /dev/sdc lvm2 — 2,00g 2,00g

Como você pode ver, ambos os dispositivos estão presentes sob a coluna PV, que significa volume físico. # Atribuir volumes físicos a grupos de volume

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias.

- **Configure o armazenamento local**
  - Listar, criar e excluir partições em discos MBR e GPT
  - Criar e remover volumes físicos
  - **Atribuir volumes físicos a grupos de volume**
  - Criar e excluir volumes lógicos
  - Configurar os sistemas para montagem de sistemas de arquivos na inicialização por ID universalmente exclusivo (UUID) ou rótulo
  - Adicionar novas partições e volumes lógicos e mudar para um sistema de forma não destrutiva

Volumes físicos para grupos de volume

Para que isso aconteça, precisamos criar um grupo de volume que possa conter um volume lógico. Poderíamos criar um "grupo" de volume que tem apenas um volume físico, mas para demonstrar a agregação de armazenamento, usaremos nossos novos volumes físicos para criar um grupo de volume chamado "dados" que tem 4GB de espaço em disco disponível.

Podemos criar um grupo de volume com o comando `vgcreate`. Tudo o que precisamos é especificar o nome do VG e os volumes físicos que serão membros do grupo de volume.

```
[root@localhost joatham]# vgcreate dados /dev/sdb /dev/sdc
  Volume group "dados" successfully created
```

Podemos listar as propriedades do nosso novo grupo de volume com `vgdisplay`.

```
[root@localhost joatham]# vgdisplay dados
  --- Volume group ---
  VG Name               dados
  System ID
  Format                lvm2
```

```
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3.99 GiB
  PE Size               4.00 MiB
  Total PE              1022
  Alloc PE / Size       0 / 0
  Free  PE / Size       1022 / 3.99 GiB
  VG UUID               CmM07M-16Ys-PZx2-XGvo-N1j3-nINX-fnIFIA
```

Observe o tamanho de `VG` 'e de cerca de `4GB`, justamente a soma dos dois volumes físicos.

> Não será exatamente a soma dos volumes, pois algum espaço é reservado para metadados.

Se verificarmos a saída do comando `pvs` novamente, podemos ver que nossos volumes físicos agora estão associados a um novo grupo de volume:

```
[root@localhost joatham]# pvs
  PV         VG    Fmt  Attr PSize   PFree
  /dev/sda2  rhel  lvm2 a--  <10,00g     0
  /dev/sdb   dados lvm2 a--   <2,00g <2,00g
  /dev/sdc   dados lvm2 a--   <2,00g <2,00g
```

Podemos ver ainda um breve resumo do próprio grupo de volume digitando:

```
[root@localhost joatham]# vgs
  VG    #PV #LV #SN Attr   VSize   VFree
  dados   2   0   0 wz--n-   3,99g 3,99g
  rhel    1   2   0 wz--n- <10,00g    0
```

Como você pode ver, nosso grupo de volume atualmente tem dois volumes físicos, zero volumes lógicos e tem a capacidade combinada dos dispositivos subjacentes.

13

Crie e exclua volumes lógicos

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias.

- **Configure o armazenamento local**
  - Listar, criar e excluir partições em discos MBR e GPT
  - Criar e remover volumes físicos
  - Atribuir volumes físicos a grupos de volume
  - **Criar e excluir volumes lógicos**
  - Configurar os sistemas para montagem de sistemas de arquivos na inicialização por ID universalmente exclusivo (UUID) ou rótulo
  - Adicionar novas partições e volumes lógicos e mudar para um sistema de forma não destrutiva

Como criar e excluir volumes lógicos

Agora que temos um grupo de volume disponível, podemos usá-lo como um pool do qual podemos alocar volumes lógicos. Ao contrário do particionamento convencional, ao trabalhar com volumes lógicos, você não precisa saber o layout do volume, pois o LVM mapeia e trata disso para você.

A instalação padrão em muitas distribuições separa o armazenamento desta forma, por ex-

emplo, os dados do usuário são mantidos no volume lógico `/home` e os logs do sistema são armazenados no volume `/var`. No caso de um usuário preencher o volume inicial, o sistema geral ainda será capaz de gravar logs e, portanto, seus serviços continuarão a ser executados enquanto o administrador do sistema pode lidar com o usuário que enlouqueceu.

Como criar um volume lógico

Vamos criar dois volumes lógicos, o primeiro denominado **banco_de_dados** com `2GB` de tamanho, o outro denominado **servidor_web** com `500MB` de espaço em disco.

Para criar volumes lógicos, usamos o comando `lvcreate`. Devemos passar o grupo de volume de onde extrair e podemos nomear o volume lógico com a opção `-n`. Para especificar o tamanho diretamente, você pode usar a opção `-L`. Se, em vez disso, você deseja especificar o tamanho em termos de número de extensões, você pode usar a opção `-l`.

```
[root@localhost 4linux]# lvcreate -L 2G -n banco_de_dados dados
  Logical volume "banco_de_dados" created.

[root@localhost 4linux]# lvcreate -L 500M -n servidor_web dados
  Logical volume "servidor_web" created.
```

Podemos exibir nossas propriedades de volume com `lvdisplay <volume>`, ou podemos listar todos os nossos volumes lógicos se não fornecermos um argumento para `lvdisplay`. Embora isso possa fornecer uma saída longa em alguns sistemas, é útil em um ambiente desconhecido, pois os caminhos dos volumes podem variar dependendo da distribuição e da versão.

```
[root@localhost 4linux]# lvdisplay /dev/dados/banco_de_dados
  --- Logical volume ---
  LV Path                /dev/dados/banco_de_dados
  LV Name                banco_de_dados
  VG Name                dados
  LV UUID                D7f9An-G0dd-kEGw-OGrP-HZlA-dQlX-yBbQbi
  LV Write Access        read/write
  LV Creation host, time rhel8rhcsa, 2019-12-28 16:53:24 +0100
  LV Status              available
  # open                 0
  LV Size                2.00 GiB
  Current LE             512
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:2

[root@localhost 4linux]# lvdisplay /dev/dados/servidor_web
  --- Logical volume ---
  LV Path                /dev/dados/servidor_web
  LV Name                servidor_web
```

```
  VG Name                dados
  LV UUID                7Ldt79-aw0i-Oydm-4d0I-JaVe-Zd8m-xKpvrD
  LV Write Access        read/write
  LV Creation host, time rhel8rhcsa, 2019-12-28 16:53:38 +0100
  LV Status              available
  # open                 0
  LV Size                500.00 MiB
  Current LE             125
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:3
```

Se verificarmos nosso grupo de volume neste ponto, podemos notar o espaço que os volumes lógicos estão usando do grupo de volume.

```
[root@localhost 4linux]# vgdisplay dados
  --- Volume group ---
  VG Name               dados
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  5
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2 # See here!
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3.99 GiB
  PE Size               4.00 MiB
  Total PE              1022
  Alloc PE / Size       637 / <2.49 GiB
  Free  PE / Size       385 / 1.50 GiB
  VG UUID               CmM07M-16Ys-PZx2-XGvo-N1j3-nINX-fnIFIA
```

Ainda podemos ver os volumes lógicos e sua relação com o grupo de volumes, selecionando a saída personalizada do comando `vgs`:

```
[root@localhost 4linux]# vgs -o +lv_size,lv_name
```

Como deletar um volume lógico

Para liberar espaço ou reorganizar volumes, podemos descartar volumes lógicos com `lvremove`.

```
[root@localhost 4linux]# lvremove /dev/dados/servidor_web
Do you really want to remove active logical volume dados/servidor_web? [y/n]: y
  Logical volume "servidor_web" successfully removed
```

Outra lista de grupo de volume mostra 500MB do volume do **servidor_web** agora foi adicionado de volta ao tamanho livre do grupo de volume.

```
shell [root@localhost 4linux]# vgdisplay dados --- Volume group --- VG Name dados System ID Format
  lvm2 Metadata Areas 2 Metadata Sequence No 6 VG Access read/write VG Status resizable MAX LV
 0 Cur LV 1 Open LV 0 Max PV 0 Cur PV 2 Act PV 2 VG Size 3.99 GiB PE Size 4.00 MiB Total PE 1022
  Alloc PE / Size 512 / 2.00 GiB Free PE / Size 510 / 1.99 GiB VG UUID CmM07M-16Ys-PZx2-XGvo-N1j3
 -nINX-fnIFIA# Configurar sistemas para montagem de sistemas de arquivos
```

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias.

- **Configure o armazenamento local**
  - Listar, criar e excluir partições em discos MBR e GPT
  - Criar e remover volumes físicos
  - Atribuir volumes físicos a grupos de volume
  - Criar e excluir volumes lógicos
  - **Configurar os sistemas para montagem de sistemas de arquivos na inicialização por ID universalmente exclusivo (UUID) ou rótulo**
  - Adicionar novas partições e volumes lógicos e mudar para um sistema de forma não destrutiva

Configurar montagens de sistemas

Em um ambiente SAN (Storage Area Network), para alta disponibilidade, um servidor pode chegar ao seu armazenamento através de vários caminhos - na verdade distribuídos e espelhados em vários discos da rede de armazenamento. Se alguns caminhos mudarem, o servidor precisará identificar o "disco" novamente. É por isso que recomendamos usar identificadores especiais definidos no dispositivo e montar por esses identificadores, não pelo nome do dispositivo, que pode mudar.

Como obter UUID de um determinado dispositivo

Para listar o UUID dos dispositivos presentes em nosso sistema, vamos abrir um terminal e usar `blkid`:

```
[root@localhost 4linux]# blkid
/dev/sda1: UUID="eef3b378-5272-45f4-ab41-97eb48bda63f" TYPE="xfs" PARTUUID="3c939719-01"
/dev/sda2: UUID="rfezEa-GlgW-jWUX-Zixs-Ydw0-EsZS-nk3JDH" TYPE="LVM2_member" PARTUUID="3
     c939719-02"
/dev/sr0: UUID="2019-10-10-18-52-14-12" LABEL="VBox_GAs_6.0.14" TYPE="iso9660"
/dev/mapper/rhel-root: UUID="9ba9c1f7-40d7-4eb2-a66b-7b27905d8011" TYPE="xfs"
/dev/mapper/rhel-swap: UUID="c08948ec-2320-4155-92d5-2c9364ccb99b" TYPE="swap"
/dev/sdb: UUID="17c1210c-8a88-42d6-b394-03f491415d5c" TYPE="ext4"
/dev/sdd: UUID="b1ce2f1a-ef90-47cd-ac50-0556d1ef12e1" TYPE="ext4"
```

Como obter e definir o rótulo de um dispositivo

Para imprimir a etiqueta do nosso dispositivo, usamos `e2label`.

```
[root@localhost 4linux]# e2label /dev/sdd
```

O mesmo utilitário é capaz de definir o rótulo. A sintaxe é `e2label <devicename> <label>`:

```
[root@localhost 4linux]#e2label /dev/sdd "disco_pequeno"
```

O utilitário `blkid`, que usamos anteriormente, também apresentará o rótulo recém-definido:

```
[root@localhost 4linux]#blkid | grep sdd
/dev/sdb: LABEL="disco_pequeno" UUID="17c1210c-8a88-42d6-b394-03f491415d5c" TYPE="ext4"
```

Como montar o dispositivo por UUID

Usar `UUID` para montar é a maneira recomendada, portanto, em uma instalação padrão do Sistema Operacional, já podemos encontrar um exemplo de como fazer isso. Se não modificamos as opções de disco na instalação, o dispositivo de inicialização provavelmente será montado pelo UUID. A configuração para montagem está no arquivo `/etc/fstab`.

```
UUID=17c1210c-8a88-42d6-b394-03f491415d5c /mnt/novo_disco ext4    defaults        0 0
```

Montar o sistema de arquivos por rótulo é praticamente a mesma coisa. Com o rótulo já definido, podemos referenciá-lo em `/etc/fstab`. Se você estiver executando esta etapa com o mesmo dispositivo, lembre-se de remover a referência UUID adicionada na etapa anterior, antes de adicionar outra que use o rótulo:

```
LABEL=disco_pequeno /mnt/novo_disco  ext4    defaults        0 0
```

Também precisaremos que o ponto de montagem exista. Portanto, vamos criar o diretório que mencionamos na entrada acima:

```
[root@localhost 4linux]# mkdir /mnt/novo_disco
```

```
[root@localhost 4linux]# mount /mnt/novo_disco/
```

Com a montagem bem-sucedida, podemos então encontrar nosso sistema de arquivos ext4 no ponto de montagem especificado. O comando `mount` sem argumentos vai listar todos os sistemas de arquivos montados. Além disso, podemos usar o `grep` para encontrar a linha na qual estamos particularmente interessados.

```
[root@localhost 4linux]# mount | grep sdd
/dev/sdd on /mnt/novo_disco type ext4 (rw,relatime,seclabel)
```

Também podemos usar o utilitário `df` para verificar o sistema de arquivos montado:

```
[root@localhost 4linux]# df -h /mnt/novo_disco/
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdd        2.0G  6.0M  1.8G   1% /mnt/novo_disco
```

Finalmente, podemos desmontar o sistema de arquivos para nos prepararmos para a próxima etapa:

```
[root@localhost 4linux]# umount /mnt/novo_disco
```

Formatando e montando os volumes lógicos existentes

Agora que temos volumes lógicos, podemos usá-los como dispositivos de bloco normais.

Os dispositivos lógicos estão disponíveis no diretório `/dev`, assim como outros dispositivos de armazenamento. Você pode acessá-los em dois lugares:

- `/dev/volume_group_name/logical_volume_name`
- `/dev/mapper/volume_group_name-logical_volume_name`

Portanto, para formatar nossos volumes lógicos com o sistema de arquivos `Ext4`, digite os comandos:

```
[root@localhost 4linux]# mkfs.ext4 /dev/mapper/dados-banco_de_dados
[root@localhost 4linux]# mkfs.ext4 /dev/mapper/dados-servidor_web
```

Após a formatação, podemos criar pontos de montagem:

```
[root@localhost 4linux]# mkdir -p /mnt/{db,www}
```

Podemos, então, montar os volumes lógicos no local apropriado:

```
[root@localhost 4linux]# mount /dev/mapper/dados-banco_de_dados /mnt/db
[root@localhost 4linux]# mount /dev/mapper/dados-servidor_web /mnt/www
```

Para tornar as montagens persistentes, adicione-as ao arquivo `/etc/fstab` da mesma forma que faria com dispositivos de bloco normais:

```
[root@localhost 4linux]# vi /etc/fstab
```

```
/etc/fstab
. . .
```

```
/dev/mapper/dados-banco_de_dados /mnt/db ext4 defaults,nofail 0 0
/dev/mapper/dados-servidor_web /mnt/www ext4 defaults,nofail 0 0
```

O sistema operacional agora deve montar os volumes lógicos LVM automaticamente na inicialização.# Adicionar novas partições e volumes lógicos e swap ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias.

- **Configure o armazenamento local**
  - Listar, criar e excluir partições em discos MBR e GPT
  - Criar e remover volumes físicos
  - Atribuir volumes físicos a grupos de volume
  - Criar e excluir volumes lógicos
  - Configurar os sistemas para montagem de sistemas de arquivos na inicialização por ID universalmente exclusivo (UUID) ou rótulo
  - **Adicionar novas partições e volumes lógicos e mudar para um sistema de forma não destrutiva**

Introducao

O **espaço de troca** ou **swap** no Linux é usado quando a quantidade de memória física (RAM) está cheia. Se o sistema precisar de mais recursos de memória e a RAM estiver cheia, as páginas inativas na memória são movidas para o espaço de troca.

Embora o espaço de troca possa ajudar máquinas com uma pequena quantidade de RAM, ele não deve ser considerado um substituto para mais RAM. O espaço de troca está localizado nos discos rígidos, que têm um tempo de acesso mais lento do que a memória física. A **Swap** pode ser uma partição de troca dedicada (recomendado), um arquivo de troca ou uma combinação de partições e arquivos de troca.

Para adicionar uma Área de Troca, você tem três opções: - Criar uma nova partição Swap; - Criar um novo arquivo de troca ou - Estender a swap de um volume lógico LVM2 existente.

No nosso caso, a última opcao será a recomendada.

Como adicionar swap

Vamos examinar nosso esquema LVM com o comando vgdisplay:

```
[root@localhost 4linux]# vgdisplay
```

```
  --- Volume group ---
  VG Name               dados
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3,99 GiB
  PE Size               4,00 MiB
  Total PE              1022
  Alloc PE / Size       637 / <2,49 GiB
  Free  PE / Size       385 / 1,50 GiB
  VG UUID               pFgqMj-Lr30-Ogay-Y6cB-muFh-9JHT-yt1dwf
```

Em seguida, usamos o comando `lvcreate` para adicionarmos `1G` de espaço que ainda restava no nosso *Volume Group* para criarmos a *Swap*:

```
[root@localhost 4linux]# lvcreate -L 1G -n swap dados
  Logical volume "swap" created.
```

Agora, com o o comando `lvdisplay`, percebemos que o espaço foi realmente alocado.

```
[root@localhost 4linux]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/dados/banco_de_dados
  LV Name                banco_de_dados
  VG Name                dados
  LV UUID                QY4eTQ-3oRO-K1zA-v1MA-VOKI-rdM8-G6Ys7j
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2021-10-05 15:20:18 -0300
  LV Status              available
  # open                 1
  LV Size                2,00 GiB
  Current LE             512
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:2

  --- Logical volume ---
  LV Path                /dev/dados/servidor_web
  LV Name                servidor_web
  VG Name                dados
  LV UUID                InRSKu-zZuP-K27y-CP0Q-5eyJ-3rpa-32WSh2
  LV Write Access        read/write
```

```
  LV Creation host, time localhost.localdomain, 2021-10-05 15:20:45 -0300
  LV Status              available
  # open                 1
  LV Size                500,00 MiB
  Current LE             125
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:3

  --- Logical volume ---
  LV Path                /dev/dados/swap
  LV Name                swap
  VG Name                dados
  LV UUID                Ba1l8Y-gFaf-Fkfg-OQit-BnIK-opA2-xcNwWu
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2021-10-05 16:58:19 -0300
  LV Status              available
  # open                 0
  LV Size                1,00 GiB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:4
```

Pronto, a criação de uma partição `swap` está a apenas um comando de distância, chamado `mkswap`!

```
[root@localhost 4linux]# mkswap /dev/mapper/dados-swap

Configurando espaço de swap versão 1, tamanho = 1024 MiB (1073737728 bytes)
nenhum rótulo, UUID=3bade52f-2507-485c-9d8e-7f35f0842ea2
```

Ative o volume lógico estendido:

```
[root@localhost 4linux]# swapon -v /dev/mapper/dados-swap

swapon: /dev/mapper/dados-swap: encontrada assinatura [tamanho de página=4096, assinatura=
     swap]
swapon: /dev/mapper/dados-swap: pagesize=4096, swapsize=1073741824, devsize=1073741824
swapon /dev/mapper/dados-swap
```

Nao se esqueça de adicionar a seguinte entrada ao arquivo `/etc/fstab`:

```
/dev/mapper/dados-swap swap swap defaults 0 0
```

14

Crie, monte, desmonte e use arquivos vfat, ext4 e xfs

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. agrupamos em várias categorias:

- **Criar e configurar sistemas de arquivos**
  - **Crie, monte, desmonte e use sistemas de arquivos vfat, ext4 e xfs**
  - Monte e desmonte sistemas de arquivos de rede usando NFS
  - Amplie os volumes lógicos existentes
  - Crie e configure diretórios set-GID para colaboração
  - Configure compactação de disco
  - Gerencie armazenamento em camadas
  - Diagnostique e corrija problemas de permissão de arquivo

Introdução

Um sistema de arquivos é um processo que controla como os dados são armazenados e recuperados de um dispositivo de armazenamento, normalmente uma unidade de disco rígido (HDD) ou unidade de estado sólido (SSD).

Ele faz a indexação de dados e também fornece uma maneira de armazenar metadados sobre esses arquivos, como suas permissões, nomes, horários de criação e modificação, além de outros atributos.

Você pode entender como organizamos nossos livros ou arquivos em uma estante. As pessoas podem ter suas próprias maneiras de organizar as coisas lá. Da mesma forma, existem muitos sistemas de arquivos para organizar e armazenar dados sobre eles.

Cada sistema operacional usa sua própria escolha de sistema de arquivos com base em seu público-alvo ou requisitos. Assim, como poucos são ricos em recursos de segurança, poucos fornecem armazenamento mais rápido e suporte a tamanhos de arquivo maiores.

Sistemas de arquivos Linux populares

Sistema de arquivos Ext4

O sistema de arquivos de quarta geração da família de sistemas de arquivos `Ext` (estendido) é o padrão no RHEL, Debian, Ubuntu e assim por diante.

Ext4 vem com alguns recursos novos e aprimorados, como:

- Metadados baseados em extensão
- Alocação atrasada
- Soma de verificação do diário
- Grande suporte de armazenamento
- Alocação multi-bloco

O tempo de reparo do sistema de arquivos (`fsck`), no `Ext4`, é muito mais rápido do que na geração anterior. Alguns reparos do sistema de arquivos demonstraram um aumento de até seis vezes no desempenho. O tamanho máximo suportado para Ext4 no RHEL 7 é 16 TB em comparação com 500 TB no XFS.

Sistema de arquivos XFS

O XFS é um sistema de arquivos com registro de 64 bits. Robusto e maduro, ele oferece suporte a arquivos muito grandes (escala para exabytes) e sistemas de arquivos em um único host. É o sistema de arquivos padrão no RHEL. O registro em **Journal** garante a integridade do sistema de arquivos após o sistema travar (por exemplo, devido a quedas de energia), mantendo um registro das operações do sistema de arquivos que pode ser reproduzido quando o sistema é reiniciado, e o sistema de arquivos remontado.

O XFS oferece suporte a uma grande variedade de recursos, incluindo o seguinte:

- Alocação atrasada
- Inodes alocados dinamicamente
- Indexação de árvore B para escalabilidade de gerenciamento de espaço livre
- Desfragmentação online
- Sistema de arquivos online crescendo
- Recursos de diagnóstico abrangentes
- Utilitários de reparo escaláveis e rápidos
- Otimizado para suportar cargas de trabalho de streaming de vídeo
- Suporta um grande número de operações paralelas
- Verificação extensiva de consistência de metadados em tempo de execução
- Algoritmos sofisticados de leitura antecipada de metadados
- Utilitários de backup e restauração totalmente integrados

O XFS é um dos sistemas de arquivos de mais alto desempenho em grandes sistemas com cargas de trabalho corporativas e uma escolha preferencial.

> Você não pode diminuir (reduzir) o tamanho dos sistemas de arquivos XFS, portanto, tome cuidado extra para não superalocar armazenamento para um sistema de arquivos existente.

Sistema de arquivos VFAT

O VFAT (Virtual File Allocation Table) é uma extensão para os sistemas de arquivos FAT16 e FAT32 incluída a partir do Windows 95 e suportada também no Linux e outros sistemas.

Inicialmente, o sistema FAT possuía uma grave limitação quanto ao tamanho dos nomes de arquivos, que não podiam ter mais que 11 caracteres, sendo 8 para o nome do arquivo e mais 3 para a extensão, como em "formular.doc". O limite de apenas 8 caracteres era um grande inconveniente para os usuários do MS-DOS. O "Boletim da 8a reunião anual de diretoria", por exemplo, teria de ser gravado na forma de algo como "8reandir.doc".

O sistema de arquivos VFAT não possui suporte a journaling. E é utilizado normalmente para transferir dados entre sistemas Windows e o Linux instalados no mesmo disco, pois pode ser lido e escrito por ambos os sistemas operacionais. O sistema de arquivos VFAT está longe de ser um sistema de arquivos utilizado para Sistemas Linux, exceto para compartinhamento/-compatibilidade entre o Windows e Linux. Se você utilizar VFAT no Linux, esteja certo que perderá alguns atributos, tais como: permissão de execução, links simbólicos, entre outras coisas.

Hands-On

Para criar uma nova partição para realizarmos nossos testes, precisamos ajustar nossa LVM previamente configurada.

Vamos criar uma nova particao chamada `sys` com um tamanho de `100 MB`. Para isso, digite:

```
[root@localhost 4linux]# lvcreate -L 100M --name sys dados
```

Crie um sistema de arquivos `ext4`

- Para criar um sistema de arquivos `ext4` chamado `/dev/dados/sys`, digite:

```
[root@localhost 4linux]# mkfs.ext4 /dev/dados/sys

[root@localhost joatham]# mkfs.ext4 /dev/dados/sys
mke2fs 1.45.6 (20-Mar-2020)
A criar sistema de ficheiros com 102400 1k blocos e 25688 inodes
UUID do sistema de ficheiros: 767fabf4-208f-4ad5-852b-61b4e9cc74ea
Seguranças de super-blocos armazenadas em blocos:
    8193, 24577, 40961, 57345, 73729

A alocar tabelas de grupo: feito
A escrever tabelas de inodes: feito
A criar diário (4096 blocos): feito
A escrever super-blocos e informação de contabilidade do sistema de ficheiros: feito
```

- Para montar este sistema de arquivos, digite:

```
[root@localhost joatham]# mount /dev/dados/sys /mnt/
```

Para montá-lo permanentemente, temos que editar o arquivo `/etc/fstab`, mas antes vamos entender como esse arquivo funciona:

O arquivo `/etc/fstab` contém uma lista de nomes de dispositivos e os diretórios nos quais os sistemas de arquivos selecionados são configurados para serem montados, bem como o tipo de sistema de arquivos e as opções de montagem. Portanto, ao montar um sistema de arquivos especificado em `/etc/fstab`, você pode escolher uma das seguintes opções:

```
mount[ opção ... ] diretório
mount[ opção ... ] dispositivo
```

Na maioria dos casos, o comando `mount` detecta o sistema de arquivos automaticamente. No entanto, existem certos sistemas de arquivos, como `NFS`(Network File System) ou `CIFS`(Common Internet File System), que não são reconhecidos e precisam ser especificados manualmente. Para especificar o tipo de sistema de arquivos, use o comando `mount` no seguinte formato:

```
mount -t type device directory
```

Modelo	Descrição
ext2	O sistema de arquivos `ext2`.
ext3	O sistema de arquivos `ext3`.
ext4	O sistema de arquivos `ext4`.
btrfs	O sistema de arquivos `btrfs`.
xfs	O sistema de arquivos `xfs`.
iso9660	O sistema de arquivos `ISO 9660`. É comumente usado por mídia ótica, normalmente CDs.
nfs	O sistema de arquivos `NFS`. É comumente usado para acessar arquivos na rede.
nfs4	O sistema de arquivos `NFSv4`. É comumente usado para acessar arquivos na rede.
udf	O sistema de arquivos `UDF`. É comumente usado por mídia ótica, normalmente DVDs.
vfat	O sistema de arquivos `FAT`. É comumente usado em máquinas que executam o sistema operacional Windows e em certas mídias digitais, como unidades flash USB ou disquetes.

Para especificar opções de montagem adicionais, use o comando da seguinte forma:

```
mount -o options device directory
```

Ao fornecer várias opções, não insira um espaço após uma vírgula ou mount interpretará incorretamente os valores após os espaços como parâmetros adicionais.

Opção	Descrição
async	Permite as operações de entrada/saída assíncronas no sistema de arquivos.
auto	Permite que o sistema de arquivos seja montado automaticamente usando o comando `mount -a`.
defaults	Fornece um alias para async,auto,dev,exec,nouser,rw,suid.
exec	Permite a execução de arquivos binários em um sistema de arquivos específico.
loop	Monta uma imagem como um dispositivo de loop.
noauto	O comportamento padrão não permite a montagem automática do sistema de arquivos usando o comando `mount -a`.
noexec	Não permite a execução de arquivos binários no sistema de arquivos específico.
nouser	Não permite que um usuário comum (ou seja, diferente de root) monte e desmonte o sistema de arquivos.
remount	Remonta o sistema de arquivos, caso já esteja montado.
ro	Monta o sistema de arquivos apenas para leitura.
rw	Monta o sistema de arquivos para leitura e gravação.
user	Permite que um usuário comum (ou seja, diferente de root) monte e desmonte o sistema de arquivos.

- Portanto, para adicionar nosso ponto de montagem, adicione a seguinte linha:

```
/dev/mapper/vg-sys /mnt ext4 defaults 1 2
```

Neste caso, o último número (aqui 2) está relacionado ao comando `fsck`: `0` significa nenhum `fsck` executado na inicialização (muito perigoso), `1` fsck é executado primeiro (sistema de arquivos raiz), `2` `fsck` é executado logo após o root sistema de arquivo.

O penúltimo argumento é em relação ao comando dump (normalmente definido como `1` para sistemas de arquivos reais, `0` para sistemas de arquivos de swap e NFS montados).

A melhor prática é executar o comando `mount -a`, cada vez que você alterar algo no arquivo `/etc/fstab` para detectar qualquer problema de inicialização antes que ele ocorra.

- Para verificar a consistência de um sistema de arquivos desmontado, digite:

```
[root@localhost joatham]# umount /mnt
```

```
[root@localhost joatham]# fsck /dev/dados/sys
fsck de util-linux 2.32.1
e2fsck 1.45.6 (20-Mar-2020)
/dev/mapper/dados-sys: limpo, 11/25688 ficheiros, 8896/102400 blocos
```

- Para obter detalhes sobre um sistema de arquivos, digite:

```
[root@localhost joatham]# fsck /dev/dados/sys
fsck de util-linux 2.32.1
e2fsck 1.45.6 (20-Mar-2020)
/dev/mapper/dados-sys: limpo, 11/25688 ficheiros, 8896/102400 blocos
[root@localhost joatham]# dumpe2fs /dev/dados/sys
dumpe2fs 1.45.6 (20-Mar-2020)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          767fabf4-208f-4ad5-852b-61b4e9cc74ea
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype extent 64bit
      flex_bg sparse_super large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              25688
Block count:              102400
Reserved block count:     5120
Free blocks:              93504
Free inodes:              25677
First block:              1
Block size:               1024
Fragment size:            1024
Group descriptor size:    64
Reserved GDT blocks:      256
Blocks per group:         8192
Fragments per group:      8192
Inodes per group:         1976
Inode blocks per group:   247
Flex block group size:    16
Filesystem created:       Tue Oct  5 18:27:06 2021
Last mount time:          Tue Oct  5 18:29:10 2021
Last write time:          Tue Oct  5 18:44:49 2021
Mount count:              1
Maximum mount count:      -1
Last checked:             Tue Oct  5 18:27:06 2021
Check interval:           0 (<none>)
Lifetime writes:          3500 kB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:           128
Journal inode:            8
Default directory hash:   half_md4
Directory Hash Seed:      b2553ce6-f91b-44a9-8970-a664c9779da4
Journal backup:           inode blocks
Checksum type:            crc32c
Checksum:                 0x35a9820c
```

```
Journal features:         journal_64bit journal_checksum_v3
Journal size:             4096k
Journal length:           4096
Journal sequence:         0x00000004
Journal start:            0
Journal checksum type:    crc32c
Journal checksum:         0xf653429d
```

Criar sistema de arquivos Xfs

- Para criar um sistema de arquivos `xfs` aqui chamado `/dev/dados/sys`, digite:

```
[root@localhost joatham]# mkfs.xfs -f /dev/dados/sys
meta-data=/dev/dados/sys         isize=512    agcount=4, agsize=6400 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=25600, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1368, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

- Para montar este sistema de arquivos, digite:

```
[root@localhost joatham]# mount /dev/dados/sys /mnt/
```

- Para montá-lo permanentemente, edite o arquivo `/etc/fstab` e adicione a seguinte linha:

```
/dev/dados/sys /mnt xfs defaults 1 2
```

- Para obter detalhes sobre um sistema de arquivos montado, digite:

```
[root@localhost joatham]# xfs_info /dev/dados/sys
meta-data=/dev/mapper/dados-sys  isize=512    agcount=4, agsize=6400 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=25600, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1368, version=2
```

```
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

- Para reparar a consistência de um sistema de arquivos desmontado, digite:

```
[root@localhost joatham]# umount /mnt
[root@localhost joatham]# xfs_repair /dev/dados/sys
Phase 1 - find and verify superblock...
Phase 2 - using internal log
        - zero log...
        - scan filesystem freespace and inode maps...
        - found root inode chunk
Phase 3 - for each AG...
        - scan and clear agi unlinked lists...
        - process known inodes and perform inode discovery...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
        - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
        - setting up duplicate extent list...
        - check for inodes claiming duplicate blocks...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
Phase 5 - rebuild AG headers and trees...
        - reset superblock...
Phase 6 - check inode connectivity...
        - resetting contents of realtime bitmap and summary inodes
        - traversing filesystem ...
        - traversal finished ...
        - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
```

Sistema de arquivos Vfat

- Para criar um sistema de arquivos `vfat` chamado `/dev/dados/sys`, digite:

```
mkfs.vfat /dev/dados/sys
mkfs.fat 4.1 (2017-01-24)
```

- Para montar este sistema de arquivos, digite:

```
[root@localhost joatham]# mount /dev/dados/sys /mnt/
```

- Para montá-lo permanentemente, edite o arquivo `/etc/fstab` e adicione a seguinte linha:

```
/dev/dados/sys /mnt vfat defaults 1 2
```

- Para reparar a consistência de um sistema de arquivos desmontado, digite:

```
[root@localhost joatham]# umount /mnt
[root@localhost joatham]# fsck.vfat /dev/dados/sys
fsck.fat 4.1 (2017-01-24)
/dev/dados/sys: 0 files, 0/51091 clusters
```

15

Montar e desmontar sistemas de arquivos de rede usando NFS

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos várias categorias:

- **Criar e configurar sistemas de arquivos**
  - Crie, monte, desmonte e use sistemas de arquivos vfat, ext4 e xfs
  - **Monte e desmonte sistemas de arquivos de rede usando NFS**
  - Amplie os volumes lógicos existentes
  - Crie e configure diretórios set-GID para colaboração
  - Configure compactação de disco
  - Gerencie armazenamento em camadas
  - Diagnostique e corrija problemas de permissão de arquivo

Introdução

Um *Network File System* (NFS) permite que hosts remotos montem sistemas de arquivos em uma rede e interajam com eles como se estivessem montados localmente. Isso permite que os administradores de sistema consolidem recursos em servidores centralizados na rede.

Como funciona o NFS

Atualmente, existem duas versões principais do NFS incluídas no Red Hat Enterprise Linux: - O NFS versão 3 (NFSv3) oferece suporte a gravações assíncronas seguras e é mais robusto no tratamento de erros do que o anterior NFSv2. Ele também oferece suporte a tamanhos e deslocamentos de arquivo de 64 bits, permitindo que os clientes acessem mais de 2 GB de dados de arquivo. - O NFS versão 4 (NFSv4) funciona por meio de firewalls e na Internet, não requer mais um serviço rpcbind, oferece suporte a ACLs e utiliza operações com estado.

O Red Hat Enterprise Linux suporta totalmente o NFS versão 4.2 (NFSv4.2) desde o lançamento do Red Hat Enterprise Linux 7.4.

A seguir, confira os recursos do NFSv4.2 no Red Hat Enterprise Linux: - Arquivos esparsos: verifica a eficiência do espaço de um arquivo e permite que o espaço reservado melhore a eficiência do armazenamento. - Reserva de espaço: permite que os servidores de armazenamento reservem espaço livre, o que impede que os servidores fiquem sem espaço. - NFS rotulado: reforça os direitos de acesso aos dados e habilita rótulos SELinux entre um cliente e um servidor para arquivos individuais em um sistema de arquivos NFS. - Aprimoramentos de layout: NFSv4.2 fornece nova operação, layoutstats()que o cliente pode usar para notificar o servidor de metadados sobre sua comunicação com o layout.

As versões do Red Hat Enterprise Linux anteriores a 7.4 suportam NFS até a versão 4.1.

A seguir estão os recursos do NFSv4.1: - Aprimora o desempenho e a segurança da rede e também inclui suporte do lado do cliente para NFS paralelo (pNFS). - Não requer mais uma conexão TCP separada para retornos de chamada, o que permite que um servidor NFS conceda delegações mesmo quando não pode entrar em contato com o cliente. Por exemplo, quando o NAT ou um firewall interfere. - Fornece exatamente uma semântica (exceto para operações de reinicialização), evitando um problema anterior pelo qual certas operações poderiam retornar um resultado impreciso se uma resposta fosse perdida e a operação fosse enviada duas vezes.

> Os clientes NFS tentam montar usando NFSv4.1 por padrão e voltam para NFSv4.0 quando o servidor não oferece suporte a NFSv4.1. A montagem, posteriormente, volta para NFSv3 quando o servidor não suporta NFSv4.0.

Serviços Requeridos

O Red Hat Enterprise Linux usa uma combinação de suporte no nível do kernel e processos daemon para fornecer compartilhamento de arquivo NFS. Todas as versões de NFS contam com Chamadas de Procedimento Remoto (RPC) entre clientes e servidores. Para compartilhar ou

montar sistemas de arquivos NFS, os seguintes serviços funcionam juntos, dependendo da versão do NFS implementada:

- **nfs** - `systemctl start nfs` inicia o servidor NFS e os processos RPC apropriados para atender a solicitações de sistemas de arquivos NFS compartilhados.
- **nfslock** - `systemctl start nfs-lock` ativa um serviço obrigatório que inicia os processos RPC apropriados, permitindo que os clientes NFS bloqueiem arquivos no servidor.
- **rpcbind** - `rpcbind` aceita reservas de porta de serviços RPC locais. Essas portas são então disponibilizadas (ou anunciadas) para que os serviços RPC remotos correspondentes possam acessá-las. `rpcbind` responde às solicitações de serviços RPC e configura conexões para o serviço RPC solicitado. Isso não é usado com NFSv4.

Os seguintes processos RPC facilitam os serviços NFS:

- **rpc.mountd** - Este processo é usado por um servidor NFS para processar MOUNTsolicitações de clientes NFSv3.
- **rpc.nfsd** - `rpc.nfsd` permite que versões e protocolos explícitos de NFS que o servidor anuncia sejam definidos.
- **lockd** - `lockd` é um thread do kernel que roda em clientes e servidores.
- **rpc.statd** - Este processo implementa o protocolo RPC Network Status Monitor (NSM), que notifica os clientes NFS quando um servidor NFS é reiniciado sem ser desligado normalmente.
- **rpc.rquotad** - Este processo fornece informações de cota de usuário para usuários remotos.
- **rpc.idmapd** - rpc.idmapdfornece upcalls de cliente e servidor NFSv4, que mapeiam entre nomes NFSv4 on-the-wire (strings na forma de) e UIDs e GIDs locais.

Sistema de arquivos de rede NFS

Para montar e desmontar sistemas de arquivos de rede NFS, você precisa configurar um servidor NFS. Instale o pacote do cliente NFS:

- Instale o pacote necessario:

```
[root@localhost joatham]# yum install nfs-utils
```

- Verifique se o serviço está em execução:

```
[root@localhost joatham]# systemctl status nfs-server●
 nfs-server.service - NFS server and services
```

```
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor preset:
        disabled)
   Active: inactive (dead)
```

- Ative o serviço:

```
[root@localhost joatham]# systemctl list-unit-files --type=service | less
```

```
[root@localhost joatham]# systemctl enable nfs-server --now
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/
     systemd/system/nfs-server.service.
```

```
[root@localhost joatham]# systemctl status nfs-server●
 nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor preset:
        disabled)
   Active: active (exited) since Tue 2021-10-05 19:47:55 -03; 18s ago
  Process: 12632 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl
       reload gssprox>
  Process: 12621 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
  Process: 12619 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
 Main PID: 12632 (code=exited, status=0/SUCCESS)

out 05 19:47:55 localhost.localdomain systemd[1]: Starting NFS server and services...
out 05 19:47:55 localhost.localdomain systemd[1]: Started NFS server and services.
```

- Precisamos da pasta e de um arquivo que será usado no compartilhamento, portanto, execute:

```
[root@localhost joatham]# mkdir /tmp/nfs_4linux
```

```
[root@localhost joatham]# vi /tmp/nfs_4linux/arquivoteste.txt
```

```
[root@localhost joatham]# cat /tmp/nfs_4linux/arquivoteste.txt
Este e meu arquivo de teste do meu servidor NFS
Talvez alguem possa ler isto
```

- Edite o arquivo `/etc/exports`

```
[root@localhost etc]# vi /etc/exports
/tmp/nfs_4linux     10.11.10.43(rw)
```

- Execute o comando para ativar o compartilhamento:

```
[root@localhost etc]# exportfs -a
```

- Na outra máquina, monte seu compartilhamento..

```
root@kali:/tmp# mount -t nfs 10.11.10.150:/nfs_4linux nfstest/
```

> Note que algo deu errado.. Precisamos arrumar as políticas de firewall Ainda teremos uma aula de `firewall-cmd`, portanto não nos aprofundaremos tanto.

- Ajustando políticas de firewall (`firewall-cmd`)

```
[root@localhost nfs_4linux]# firewall-cmd --list-services
cockpit dhcpv6-client ssh
```

```
[root@localhost nfs_4linux]# firewall-cmd --add-service=nfs
success
```

- Tente novamente o compartilhamento

```
root@kali:/tmp# mount -t nfs 10.11.10.150:/nfs_4linux nfstest/
```

> Agora rodou!

- Visualizando o compartilhamento

```
root@kali:/tmp# df -h
Filesystem                Size  Used Avail Use% Mounted on
udev                      966M     0  966M   0% /dev
tmpfs                     200M  988K  199M   1% /run
/dev/sda1                  78G   12G   62G  16% /
tmpfs                     996M   40M  957M   4% /dev/shm
tmpfs                     5.0M     0  5.0M   0% /run/lock
tmpfs                     996M     0  996M   0% /sys/fs/cgroup
tmpfs                     200M   12K  200M   1% /run/user/1000
10.11.10.150:/nfs_4linux  8.9G  5.5G  3.5G  61% /tmp/nfstest
```

- Ajustando permissão do compartilhamento

```
root@kali:/tmp# umount nfstest/
```

```
[root@localhost nfs_4linux]# vi /etc/exports
/tmp/nfs_4linux     10.11.10.43(rw,no_root_squash)
```

```
[root@localhost nfs_4linux]# exportfs -a
```

```
root@kali:/tmp# mount -t nfs 10.11.10.150:/nfs_4linux nfstest/
```

16

Amplie os volumes lógicos existentes

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos várias categorias:

- **Criar e configurar sistemas de arquivos**
  - Crie, monte, desmonte e use sistemas de arquivos vfat, ext4 e xfs
  - Monte e desmonte sistemas de arquivos de rede usando NFS
  - **Amplie os volumes lógicos existentes**
  - Crie e configure diretórios set-GID para colaboração
  - Configure compactação de disco
  - Gerencie armazenamento em camadas
  - Diagnostique e corrija problemas de permissão de arquivo

Introdução

Antes de estender qualquer volume lógico não criptografado, vamos criar mais um de `100MB` chamado `extensao` no grupo de volume `dados`:

```
[root@localhost nfs_4linux]# vgs
```

```
  VG    #PV #LV #SN Attr   VSize   VFree
  dados   2   4   0 wz--n-   3,99g 416,00m
  rhel    1   2   0 wz--n- <10,00g      0
```

```
[root@localhost nfs_4linux]# lvcreate -L 100M -n extensao dados
  Logical volume "extensao" created.
```

```
[root@localhost nfs_4linux]# lvdisplay /dev/dados/extensao
  --- Logical volume ---
  LV Path                /dev/dados/extensao
  LV Name                extensao
  VG Name                dados
  LV UUID                EYrRNI-G3xK-8AWW-dS60-iIOZ-FzeQ-aSwWo1
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2021-10-05 21:12:45 -0300
  LV Status              available
  # open                 0
  LV Size                100,00 MiB
  Current LE             25
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:6
```

Os comandos `lvextend` e `lvreduce` são usados para, respectivamente, aumentar ou diminuir o tamanho de um volume lógico. Por questões de brevidade, o comando `lvresize` não é discutido aqui, mas as mesmas operações podem ser feitas de uma maneira ligeiramente diferente com este comando.

Extensão de volume lógico Ext4

Para criar um sistema de arquivos `ext4` no volume lógico criado anteriormente, digite:

```
[root@localhost nfs_4linux]# mkfs.ext4 /dev/dados/extensao
mke2fs 1.45.6 (20-Mar-2020)
A criar sistema de ficheiros com 102400 1k blocos e 25688 inodes
UUID do sistema de ficheiros: 54ace52e-de2f-4cd0-ba39-524a2242a6bf
Seguranças de super-blocos armazenadas em blocos:
    8193, 24577, 40961, 57345, 73729

A alocar tabelas de grupo: feito
A escrever tabelas de inodes: feito
A criar diário (4096 blocos): feito
A escrever super-blocos e informação de contabilidade do sistema de ficheiros: feito
```

```
[root@localhost nfs_4linux]# lvextend --size +50M -r /dev/dados/extensao
  Rounding size to boundary between physical extents: 52,00 MiB.
fsck de util-linux 2.32.1
/dev/mapper/dados-extensao: limpo, 11/25688 ficheiros, 8896/102400 blocos
  Size of logical volume dados/extensao changed from 100,00 MiB (25 extents) to 152,00 MiB
      (38 extents).
  Logical volume dados/extensao successfully resized.
resize2fs 1.45.6 (20-Mar-2020)
A redimensionar o sistema de ficheiros em /dev/mapper/dados-extensao para 155648 (1k)
    blocos.
O sistema de ficheiros em /dev/mapper/dados-extensao tem agora 155648 (1k) blocos.
```

```
[root@localhost nfs_4linux]# lvdisplay /dev/dados/extensao
  --- Logical volume ---
  LV Path                /dev/dados/extensao
  LV Name                extensao
  VG Name                dados
  LV UUID                EYrRNI-G3xK-8AWW-dS60-iIOZ-FzeQ-aSwWo1
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2021-10-05 21:12:45 -0300
  LV Status              available
  # open                 0
  LV Size                152,00 MiB
  Current LE             38
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:6
```

Finalmente, monte o novo sistema de arquivos em `/mnt`:

```
[root@localhost nfs_4linux]# mount /dev/dados/extensao /mnt/
[root@localhost nfs_4linux]# df -h
Sist. Arq.                 Tam. Usado Disp. Uso% Montado em
devtmpfs                   376M     0  376M   0% /dev
tmpfs                      405M     0  405M   0% /dev/shm
tmpfs                      405M  6,4M  399M   2% /run
tmpfs                      405M     0  405M   0% /sys/fs/cgroup
/dev/mapper/rhel-root      8,9G  5,5G  3,5G  61% /
/dev/sda1                 1014M  323M  692M  32% /boot
tmpfs                       81M  4,6M   77M   6% /run/user/1000
/dev/sr0                    59M   59M     0 100% /run/media/joatham/VBox_GAs_6.1.18
/dev/mapper/dados-extensao  144M  1,6M  132M   2% /mnt
```

> Para alocar todo o espaço disponível no grupo de volume para um volume lógico `/dev/dados/extensao`, formatado em `Ext4`, digite :

```
[root@localhost nfs_4linux]# lvextend -l +100%FREE -r /dev/dados/extensao
```

Redução de volume lógico Ext4

Por outro lado, para reduzir o tamanho de um volume lógico em 50 MB, você deve seguir estas etapas:

Primeiro, desmonte o sistema de arquivos `/dev/dados/extensao`:

```
[root@localhost nfs_4linux]# umount /dev/dados/extensao
```

Reduza o tamanho do volume lógico `/dev/dados/extensao` e do sistema de arquivos associado ao mesmo tempo `-r`:

```
[root@localhost nfs_4linux]# lvreduce --size -50M -r /dev/dados/extensao
  Rounding size to boundary between physical extents: 48,00 MiB.
fsck de util-linux 2.32.1
/dev/mapper/dados-extensao: 11/37544 ficheiros (0.0% não-contíguos), 10390/155648 blocos
resize2fs 1.45.6 (20-Mar-2020)
A redimensionar o sistema de ficheiros em /dev/mapper/dados-extensao para 106496 (1k)
     blocos.
O sistema de ficheiros em /dev/mapper/dados-extensao tem agora 106496 (1k) blocos.

  Size of logical volume dados/extensao changed from 152,00 MiB (38 extents) to 104,00 MiB
       (26 extents).
  Logical volume dados/extensao successfully resized.
```

```
[root@localhost nfs_4linux]# lvdisplay /dev/dados/extensao
  --- Logical volume ---
  LV Path                /dev/dados/extensao
  LV Name                extensao
  VG Name                dados
  LV UUID                EYrRNI-G3xK-8AWW-dS60-iIOZ-FzeQ-aSwWo1
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2021-10-05 21:12:45 -0300
  LV Status              available
  # open                 0
  LV Size                104,00 MiB
  Current LE             26
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:6
```

Monte novamente o sistema de arquivos `/dev/dados/extensao`:

```
[root@localhost nfs_4linux]# mount /dev/dados/extensao /mnt/
[root@localhost nfs_4linux]# df -h
Sist. Arq.                 Tam. Usado Disp. Uso% Montado em
devtmpfs                   376M     0  376M   0% /dev
tmpfs                      405M     0  405M   0% /dev/shm
tmpfs                      405M  6,4M  399M   2% /run
tmpfs                      405M     0  405M   0% /sys/fs/cgroup
/dev/mapper/rhel-root      8,9G  5,5G  3,5G  61% /
/dev/sda1                 1014M  323M  692M  32% /boot
tmpfs                       81M  4,6M   77M   6% /run/user/1000
/dev/sr0                    59M   59M     0 100% /run/media/joatham/VBox_GAs_6.1.18
/dev/mapper/dados-extensao  97M  1,6M   89M   2% /mnt
```

Extensão de volume lógico XFS

Para criar um sistema de arquivos XFS no volume lógico criado anteriormente, chamado `extensao`, digite:

```
[root@localhost nfs_4linux]# mkfs.xfs -f /dev/dados/extensao
meta-data=/dev/dados/extensao    isize=512    agcount=4, agsize=6656 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=26624, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1368, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

```
[root@localhost nfs_4linux]# lvextend --size +50M -r /dev/dados/extensao
  Rounding size to boundary between physical extents: 52,00 MiB.
Phase 1 - find and verify superblock...
Phase 2 - using internal log
        - zero log...
        - scan filesystem freespace and inode maps...
        - found root inode chunk
Phase 3 - for each AG...
        - scan (but don't clear) agi unlinked lists...
        - process known inodes and perform inode discovery...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
        - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
        - setting up duplicate extent list...
        - check for inodes claiming duplicate blocks...
        - agno = 0
```

```
        - agno = 1
        - agno = 2
        - agno = 3
No modify flag set, skipping phase 5
Phase 6 - check inode connectivity...
        - traversing filesystem ...
        - traversal finished ...
        - moving disconnected inodes to lost+found ...
Phase 7 - verify link counts...
No modify flag set, skipping filesystem flush and exiting.
  Size of logical volume dados/extensao changed from 104,00 MiB (26 extents) to 156,00 MiB
       (39 extents).
  Logical volume dados/extensao successfully resized.
meta-data=/dev/mapper/dados-extensao isize=512    agcount=4, agsize=6656 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=26624, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1368, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 26624 to 39936
```

```
[root@localhost nfs_4linux]# lvdisplay /dev/dados/extensao
  --- Logical volume ---
  LV Path                /dev/dados/extensao
  LV Name                extensao
  VG Name                dados
  LV UUID                EYrRNI-G3xK-8AWW-dS60-iIOZ-FzeQ-aSwWo1
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2021-10-05 21:12:45 -0300
  LV Status              available
  # open                 0
  LV Size                156,00 MiB
  Current LE             39
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:6
```

```
shell [root@localhost nfs_4linux]# df -h Sist. Arq. Tam. Usado Disp. Uso% Montado em devtmpfs 376
 M 0 376M 0% /dev tmpfs 405M 0 405M 0% /dev/shm tmpfs 405M 6,4M 399M 2% /run tmpfs 405M 0 405M
 0% /sys/fs/cgroup /dev/mapper/rhel-root 8,9G 5,5G 3,5G 61% / /dev/sda1 1014M 323M 692M 32% /boot
  tmpfs 81M 4,6M 77M 6% /run/user/1000 /dev/sr0 59M 59M 0 100% /run/media/joatham/VBox_GAs_6.1.18
  /dev/mapper/dados-extensao 151M 9,3M 142M 7% /mnt# Criar e configurar diretórios set-GID
```

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos várias categorias:

- **Criar e configurar sistemas de arquivos**
  - Crie, monte, desmonte e use sistemas de arquivos vfat, ext4 e xfs
  - Monte e desmonte sistemas de arquivos de rede usando NFS
  - Amplie os volumes lógicos existentes
  - **Crie e configure diretórios set-GID para colaboração**
  - Configure compactação de disco
  - Gerencie armazenamento em camadas
  - Diagnostique e corrija problemas de permissão de arquivo

Introdução

As três permissões básicas do linux (rwx) não dão toda flexibilidade para controlar acesso aos arquivos e diretórios existentes no Sistema Operacional.

Por isso, o Linux conta com mais três modelos especiais para controle de acesso, chamados SUID (Set User id), SGID (Set Group id) e Sticky (Sticky bit).

Veja um exemplo de permissão especial na figura abaixo:

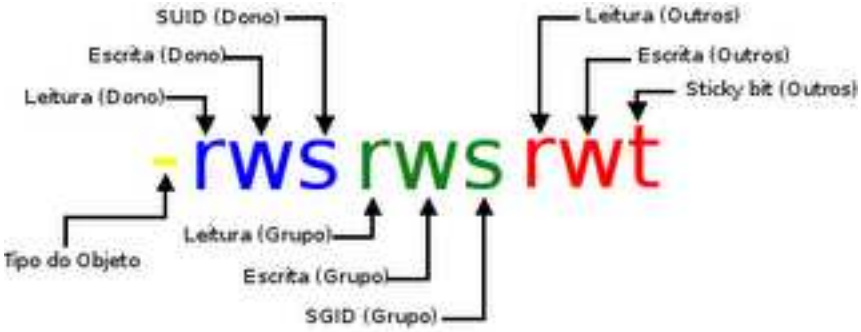


Fig. 16.1: SIGD

SGID (Set Group ID)

A propriedade SGID tem a mesma função que o SUID para arquivos executáveis. No entanto, a propriedade SGID tem um efeito especial para os diretórios.

Quando SGID é aplicado em um diretório, os novos arquivos que são criados dentro dele assumem o mesmo ID de Grupo do diretório com a propriedade SGID aplicado.

A permissão de acesso especial SGID pode aparecer somente no campo Grupo.

Exemplo:

Se no diretório /home/joatham tem o grupo 4linux e tem o SGID habilitado, então todos os arquivos dentro do diretório /home/joatham serão criados com o grupo 4linux.

Este é um importante atributo para uma security, assumindo que todos os arquivos compartilhados devem ter o mesmo grupo.

- Aplicando SGID:

Aplicando a propriedade SGID em um diretório executável, utilizando formato simbólico (s):

```
[root@localhost joatham]# chmod g+s /home/security
[root@localhost joatham]# ls -lah /home/security
drwxr-sr-x 2 joatham security 48 2021-09-26 23:21 .
```

Aplicando a propriedade SGID em um diretório executável, utilizando formato octal (2):

```
[root@localhost joatham]# chmod 2750 /home/security
[root@localhost joatham]# ls -lah /home/security
drwxr-s--- 2 joatham security 48 2021-09-26 23:21 .
```

- Retirando SGID:

```
[root@localhost joatham]# chmod g-s /home/security
[root@localhost joatham]# ls -lah /home/security
drwxr-xr-x 2 joatham security 48 2021-09-26 23:21 .
```

- Procurando SGID:

Procurando a propriedade SGID em um diretório executável, utilizando formato simbólico (s):

```
[root@localhost joatham]# find /home -perm /g=s
/home/security
```

- Procurando a propriedade SUID em um diretório executável, utilizando formato octal (2):

```
[root@localhost joatham]# find /home -perm -2000
/home/security
```

17

Configurar compactação de disco

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos várias categorias:

- **Criar e configurar sistemas de arquivos**
  - Crie, monte, desmonte e use sistemas de arquivos vfat, ext4 e xfs
  - Monte e desmonte sistemas de arquivos de rede usando NFS
  - Amplie os volumes lógicos existentes
  - Crie e configure diretórios set-GID para colaboração
  - **Configure compactação de disco**
  - Gerencie armazenamento em camadas
  - Diagnostique e corrija problemas de permissão de arquivo

Introdução

Virtual Data Optimizer (VDO) é um módulo de mapeamento de dispositivo que adiciona recursos de redução de dados à pilha de armazenamento de bloco do Linux. O VDO usa técnicas de compactação em linha e desduplicação de dados para reduzir de forma transparente os dados à medida que são gravados na mídia de armazenamento.

O VDO combina três técnicas: - Eliminação de bloco zero - Funciona eliminando blocos de dados consistindo inteiramente de zeros; - Desduplicação de dados - Elimina cópias idênticas

de blocos de dados que já foram armazenados, para reduzir a pegada de dados; - Compactação de dados - Finalmente, a compactação de dados é aplicada, o que reduz o tamanho dos blocos exclusivos de dados armazenados.

Ao utilizar essas técnicas, o VDO pode aumentar drasticamente a eficiência de armazenamento e utilização da largura de banda da rede.

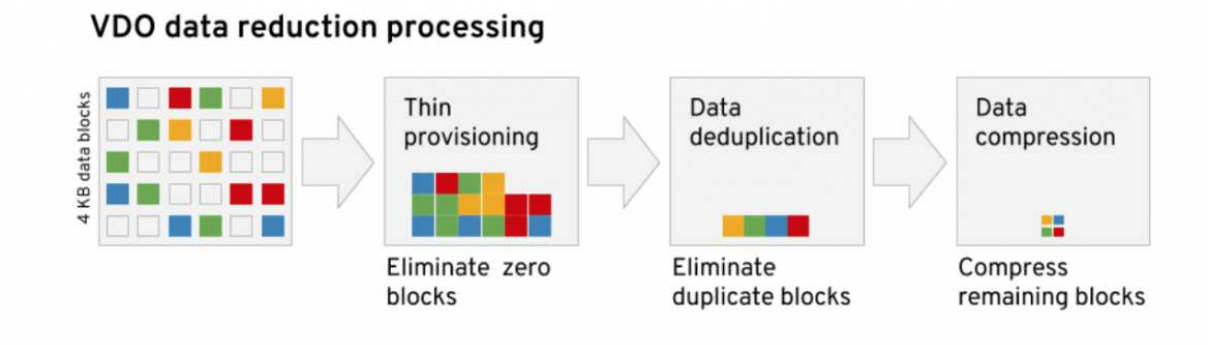


Fig. 17.1: VDO1

Por que o VDO é importante?

O VDO pode ser usado para economizar espaço de armazenamento e reduzir custos. Por ser um recurso do Red Hat Enterprise Linux (RHEL), ele pode ser usado em qualquer lugar que o RHEL seja implantado; economias semelhantes podem ser vistas tanto no data center tradicional quanto nas implantações baseadas em nuvem.

No data center tradicional, isso significa que você pode reaproveitar os recursos de armazenamento que já possui e fazer um uso mais eficiente dos equipamentos futuros. A replicação de dados corporativos também pode se beneficiar dessa eficiência, pois menos dados no armazenamento significa menos dados para replicar.

Na nuvem, o VDO também permite cortar custos de armazenamento. Dependendo de suas necessidades de implantação, isso pode se traduzir em custos mais baixos por instância de computação, custos mais baixos para armazenamento em bloco externo baseado em nuvem e custos reduzidos para retenção de longo prazo de instantâneos de dados. Além disso, a pegada reduzida no local ou na nuvem se traduz em requisitos de largura de banda reduzidos para copiar os dados de ou para a nuvem ou mesmo entre as nuvens.

A quantidade de redução de dados que você verá com o VDO pode variar dependendo dos tipos de dados armazenados e do fluxo de trabalho que os cria e armazena. Os tipos de dados já compactados, como arquivos de vídeo ou áudio, não se beneficiarão com essa tecnologia, mas backups online, máquina virtual e implantações de contêiner verão benefícios substanciais. Não é incomum que os usuários relatem taxas de redução de dados 6:1 em ambientes mistos

de contêineres e VMs usando tecnologias de desduplicação e compactação, como as fornecidas pela VDO. Vários bons candidatos para redução de dados com VDO estão listados abaixo.

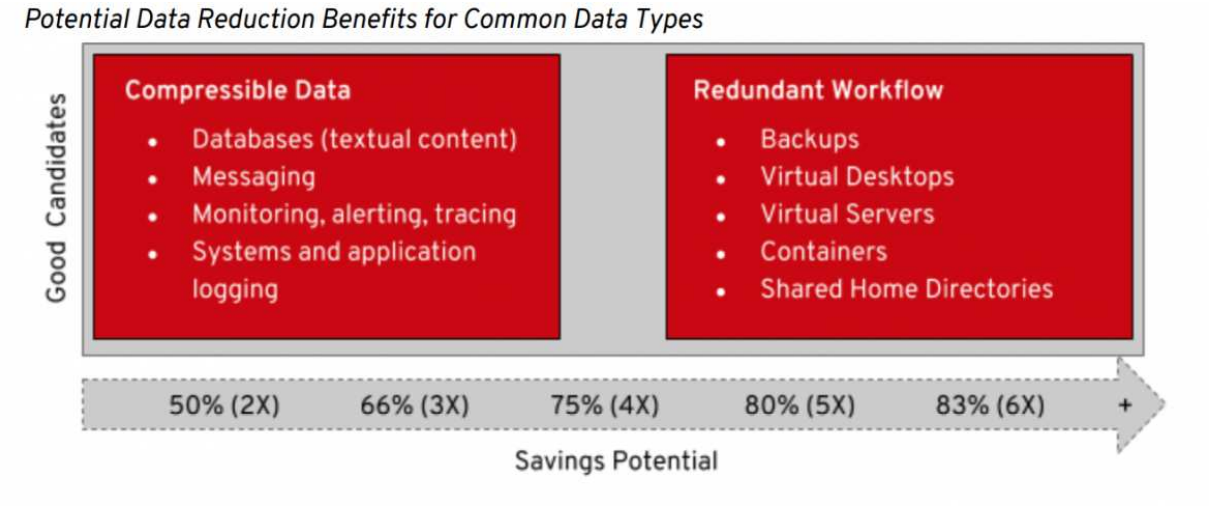


Fig. 17.2: VDO2

Como faço para usar o VDO?

O VDO opera na camada de bloco do Linux. Isso permite que ele forneça benefícios para o armazenamento local, bem como para soluções de armazenamento distribuído de blocos, arquivos ou objetos.

Modelos de implantação VDO

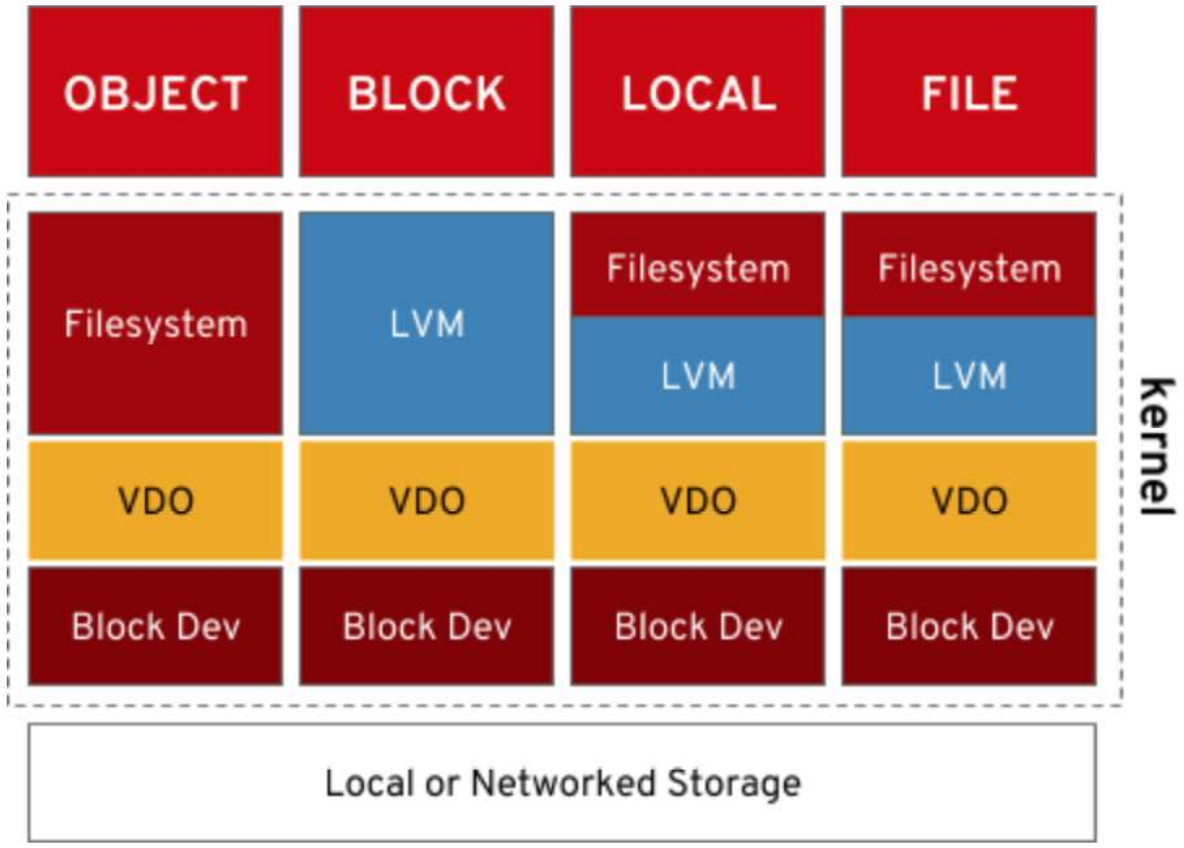


Fig. 17.3: VDO3

O VDO fica em cima de um dispositivo de armazenamento em bloco existente que pode ser qualquer coisa, desde um único disco local a um dispositivo RAID ou até mesmo um LVM de um array de armazenamento corporativo. Recursos como criptografia e RAID de software ficam abaixo do VDO, enquanto componentes como LVM e sistemas de arquivos são dispostos em camadas sobre um dispositivo VDO.

A configuração do VDO para este caso é simples e pode ser feita no RHEL usando o utilitário de linha de comando vdo ou integrando o módulo Ansible fornecido em seus manuais que executam a configuração de armazenamento.

Para configurar o VDO, você precisa saber o seguinte:

- O nome do dispositivo de bloco subjacente (opção do dispositivo)
- O nome do dispositivo de bloco otimizado que o VDO apresentará (opção de nome)
- O tamanho lógico que você deseja apresentar às camadas de armazenamento acima do VDO (opção vdoLogicalSize) Se você não apresentar este último parâmetro, o VDO criará um volume que fornece um mapeamento 1:1 entre os blocos físicos e lógicos.

Requisitos e recomendações

- **Memória** - Cada volume VDO tem dois requisitos de memória distintos:
- **O módulo VDO** - O VDO requer 370MB de RAM mais 268MB adicionais para cada 1 TB de armazenamento físico gerenciado pelo volume.
- **O índice UDS (Universal Deduplication Service).** - O UDS requer um mínimo de 250 MB de RAM, que também é a quantidade padrão usada pela desduplicação. A memória necessária para o índice UDS é determinada pelo tipo de índice e pelo tamanho necessário da janela de desduplicação.

Tipo do index	Janela de Desduplicacao	Anotacoes
Dense	1TB por 1GB de RAM	Um índice *dense* de 1GB geralmente é suficiente para até 4TB de armazenamento físico
Sparse	10TB por 1GB de RAM	Um índice *sparse* de 1GB geralmente é suficiente para até 40TB de armazenamento físico

> Sparse é a configuração recomendada.

Storage

Logical Size

Especifica o tamanho do volume VDO lógico. O tamanho lógico do VDO é a quantidade de armazenamento que informamos ao sistema operacional que temos. Devido à redução e desduplicação, esse número será maior do que o tamanho físico real. Essa proporção vai variar de acordo com o tipo de dados que está sendo armazenado (binário, vídeo, áudio, dados compactados terão uma proporção muito baixa). Use um tamanho lógico que seja dez vezes o tamanho físico do seu dispositivo de bloco. Por exemplo, se o seu dispositivo de bloco tiver 1TB de tamanho, use 10TB aqui.

Para armazenamento de objetos, use um tamanho lógico três vezes maior que o tamanho

físico do seu dispositivo de bloco. Por exemplo, se o seu dispositivo de bloco tiver 1 TB de tamanho, use `3TB` aqui.

Slab Size

Especifica o tamanho do incremento pelo qual um `VDO` é aumentado. Todas as placas para um determinado volume terão o mesmo tamanho, que pode ser qualquer potência de 2 múltiplos de 128 MB até 32 GB. Pelo menos uma placa inteira é reservada pelo VDO para metadados e, portanto, não pode ser usada para armazenar dados do usuário.

O tamanho da placa padrão é `2GB` para facilitar a avaliação do VDO em sistemas de teste menores. Um único volume `VDO` pode ter até 8096 placas. Portanto, na configuração padrão com placas de 2 GB, o armazenamento físico máximo permitido é de `16TB`. Ao usar placas de `32GB`, o armazenamento físico máximo permitido é `256TB`.

Tamanho do volume fisico	Recomendado Slab
`10-99GB`	`1GB`
`100GB - 1TB`	`2GB`
`2-256TB`	`32GB`

Exemplos de requisitos de sistema VDO por tamanho de volume físico

As tabelas a seguir fornecem requisitos de sistema aproximados de `VDO` com base no tamanho do volume físico subjacente. Cada tabela lista os requisitos apropriados para a implantação pretendida, como armazenamento primário ou armazenamento de backup.

Tamanho fisico do volume	Uso da RAM	Uso do Disco	Tipo do index
10GB-1TB	250MB	2.5GB	Dense
2-10TB	1GB	10GB	Dense
2-10TB	250MB	22GB	Sparse
11-50TB	2GB	170GB	Sparse
51-100TB	3GB	255GB	Sparse
101-256TB	12GB	1020GB	Sparse

Hands On

Etapa 1: instalar o Virtual Data Optimizer (VDO)

Para distribuição RHEL e CentOS Linux, você pode instalar facilmente o mapeador de dispositivo Linux Virtual Data Optimizer (VDO) executando os comandos abaixo.

```
root@localhost joatham]# yum install vdo kmod-kvdo
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 20:27:58 atrás em ter 05 out 2021
     18:09:59 -03.
O pacote vdo-6.2.4.14-14.el8.x86_64 já está instalado.
O pacote kmod-kvdo-6.2.4.26-77.el8.x86_64 já está instalado.
Dependências resolvidas.
Nada para fazer.
Concluído!
```

Etapa 2: Verifique o funcionamento do serviço

Após a instalação bem-sucedida, inicie, ative e verifique o daemon `vdo`. Por padrão, ela já fica ativada, mas não custa nada verificar.

```
systemctl start vdo
$ sudo systemctl enable vdo
$ sudo systemctl status vdo
```

Etapa 3: Criação do volume VDO

Volumes `VDO` são os dispositivos lógicos que você cria usando `VDO`. Eles são tratados como partições de disco. Você apenas os formata com um sistema de arquivos e, em seguida, um volume `VDO` pode ser montado como um sistema de arquivos normal. No nosso caso, podemos usar nossa LVM (/dev/mapper/dados) já criada ou adicionamos um novo disco para realizar a instalação.

Neste caso, adicionei um disco de `8GB` que será usado para esse procedimento.

```
[root@localhost joatham]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                   8:0    0   11G  0 disk
 |--sda1                8:1    0    1G  0 part /boot
  `-sda2                8:2    0   10G  0 part
   |--rhel-root        253:0    0  8,9G  0 lvm  /
    `-rhel-swap        253:1    0  1,1G  0 lvm  [SWAP]
sdb                   8:16   0    2G  0 disk
```

```
  `-dados-banco_de_dados
                    253:2    0    2G  0 lvm  /mnt/db
sdc                   8:32   0    2G  0 disk
 |--dados-banco_de_dados
 |                  253:2    0    2G  0 lvm  /mnt/db
 |--dados-servidor_web 253:3    0  500M  0 lvm  /mnt/www
 |--dados-swap       253:4    0    1G  0 lvm
 |--dados-sys        253:5    0  100M  0 lvm
 `-dados-extensao    253:6    0  156M  0 lvm
sdd                   8:48   0    8G  0 disk
sr0                  11:0    1 58,3M  0 rom  /run/media/joatham/VBox_GAs_6.1.18
```

É assim que você criará um volume VDO:

```
[root@localhost joatham]# vdo create --name 4linuxvdo --device /dev/sdd --vdoLogicalSize 5G
Creating VDO 4linuxvdo
      The VDO volume can address 4 GB in 2 data slabs, each 2 GB.
      It can grow to address at most 16 TB of physical storage in 8192 slabs.
      If a larger maximum size might be needed, use bigger slabs.
Starting VDO 4linuxvdo
Starting compression on VDO 4linuxvdo
VDO instance 0 volume is ready at /dev/mapper/4linuxvdo
```

Onde:

- **4linuxvdo** - é o nome do dispositivo lógico que o VDO apresenta ao usuário.
- **/dev/sdd** - é um dispositivo de bloco a ser usado pelo volume VDO.
- **5G** - é o tamanho lógico do volume VDO. Isso é opcional e pode ser maior do que o tamanho físico do dispositivo de bloco real.

Exiba uma lista de volumes iniciados e não iniciados.

```
[root@localhost joatham]# vdo list --all
4linuxvdo
```

Execute o comando `vdo status` para analisar o volume.

```
[root@localhost joatham]# vdo status -n 4linuxvdo
VDO status:
  Date: '2021-10-06 14:54:40-03:00'
  Node: localhost.localdomain
Kernel module:
  Loaded: true
  Name: kvdo
  Version information:
    kvdo version: 6.2.4.26
```

```
Configuration:
  File: /etc/vdoconf.yml
  Last modified: '2021-10-06 14:48:13'
VDOs:
  4linuxvdo:
    Acknowledgement threads: 1
    Activate: enabled
    Bio rotation interval: 64
    Bio submission threads: 4
    Block map cache size: 128M
    Block map period: 16380
    Block size: 4096
    CPU-work threads: 2
    Compression: enabled
    Configured write policy: auto
    Deduplication: enabled
```

De uma forma mais fácil, verifique se a compactação e a desduplicação foram ativadas.

```
[root@localhost joatham]# vdo status -n 4linuxvdo | egrep 'Compression|Deduplication'
    Compression: enabled
    Deduplication: enabled
```

Você pode aumentar um volume existente com o comando `vdo growLogical`. Vamos aumentar o volume para `8GB` de capacidade total.

```
[root@localhost joatham]# vdo status -n 4linuxvdo | grep size
    Block map cache size: 128M
    Block size: 4096
    Logical size: 8G
    Max discard size: 4K
    Physical size: 8G
    Slab size: 2G
        block map cache size: 134217728
        block size: 4096
```

Etapa 4: Formatar o volume VDO com um sistema de arquivos.

Você pode formatar o volume `VDO` com um tipo de sistema de arquivos de sua escolha ou criar um `PV`, `VG` e `LV` a partir dele.

```
[root@localhost joatham]# mkfs.xfs /dev/mapper/4linuxvdo
```

Para criacao de `LVM`

```
[root@localhost joatham]# pvcreate /dev/mapper/4linuxvdo
  Physical volume "/dev/mapper/4linuxvdo" successfully created.
```

```
[root@localhost joatham]# vgcreate vg4linuxvdo /dev/mapper/4linuxvdo
  Volume group "vg4linuxvdo" successfully created
```

```
[root@localhost joatham]# lvcreate -n lv4linuxvdo -l+100%FREE vg4linuxvdo
  Logical volume "lv4linuxvdo" created.
```

```
[root@localhost joatham]# mkfs -t xfs /dev/mapper/vg4linuxvdo-lv4linuxvdo
meta-data=/dev/mapper/vg4linuxvdo-lv4linuxvdo isize=512    agcount=4, agsize=524032 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=2096128, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
Discarding blocks...Done.
```

Agora você pode registrar um novo dispositivo e montá-lo.

```
[root@localhost joatham]# udevadm settle
[root@localhost joatham]# mkdir /4linuxvdo
[root@localhost joatham]# mount /dev/mapper/vg4linuxvdo-lv4linuxvdo /4linuxvdo/
```

> Lembre-se: para montagem persistente, edite o arquivo `/etc/fstab`.

> UUID=XXXXX /4linuxvdo xfs defaults,x-systemd.requires=vdo.service 0 0.

Você também pode exibir estatísticas em formato legível.

```
[root@localhost joatham]# vdostats --hu
Device                    Size      Used Available Use% Space saving%
/dev/mapper/4linuxvdo      8.0G      4.0G      4.0G  50%           98%
```

Etapa 5: Teste de desduplicação

Baixaremos um arquivo ISO para testar a desduplicação.

```
[root@localhost joatham]# wget http://mirror.centos.org/centos/7/os/x86_64/images/boot.iso
--2021-10-06 15:26:39--  http://mirror.centos.org/centos/7/os/x86_64/images/boot.iso
Resolvendo mirror.centos.org (mirror.centos.org)... 187.45.180.2, 2804:9c4:2f:6::10
Conectando-se a mirror.centos.org (mirror.centos.org)|187.45.180.2|:80... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 602931200 (575M) [application/octet-stream]
Salvando em: “boot.”iso

boot.iso             100%[===================>] 575,00M  2,04MB/s    em 4m 0s

2021-10-06 15:30:39 (2,39 MB/s) - “boot.”iso salvo [602931200/602931200]
```

Copie o arquivo para o diretório `/4linuxvdo`.

```
[root@localhost joatham]# cp boot.iso /4linuxvdo/
```

Verifique as estatísticas de armazenamento.

```
[root@localhost joatham]# vdostats --hu
Device                    Size      Used Available Use% Space saving%
/dev/mapper/4linuxvdo      8.0G      4.6G      3.4G  57%            2%
```

Você pode notar que o valor do campo Usado aumentou de `4,0G` para `4,6G` porque copiamos um arquivo para o volume que ocupa algum espaço.

Vamos fazer uma segunda cópia do mesmo arquivo.

```
[root@localhost joatham]# cp boot.iso /4linuxvdo/boot2.iso
```

Visualize as estatísticas de volume novamente.

```
[root@localhost joatham]# vdostats --hu
Device                    Size      Used Available Use% Space saving%
```

```
/dev/mapper/4linuxvdo      8.0G      4.6G      3.4G  57%           50%
```

Você pode ver que o espaço do volume usado não mudou. Em vez disso, a porcentagem do espaço do volume salvo aumentou para `50%`, provando que a desduplicação de dados ocorreu para reduzir o consumo de espaço para as cópias redundantes do mesmo arquivo.

18

Gerenciar armazenamento em camadas

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos várias categorias:

- **Criar e configurar sistemas de arquivos**
  - Crie, monte, desmonte e use sistemas de arquivos vfat, ext4 e xfs
  - Monte e desmonte sistemas de arquivos de rede usando NFS
  - Amplie os volumes lógicos existentes
  - Crie e configure diretórios set-GID para colaboração
  - Configure compactação de disco
  - **Gerencie armazenamento em camadas**
  - Diagnostique e corrija problemas de permissão de arquivo

Conhecendo o gerenciador de armazenamentos `Stratis`

`Stratis` é um dos novos recursos que vem com a distribuição RHEL. Essa solução de gerenciamento de armazenamento local se concentra na simplicidade e na usabilidade aprimorada, ao mesmo tempo que fornece acesso a recursos avançados de armazenamento como: 1. Gestão baseada em pool 2. Provisionamento fino 3. Snapshots do sistema de arquivos 4. Monitoramento

`Stratisd` é o daemon por trás do `Stratis`. Ele ajuda a configurar os componentes de armazenamento no sistema de arquivos `XFS` e no subsistema do mapeador de dispositivos.

No momento, `Stratis` suporta volumes lógicos `LVM`, discos rígidos, SSDs, NVMe e dispositivos de armazenamento `ISCIs`.

Antes de prosseguirmos, vamos dar uma olhada em alguns dos termos técnicos que você provavelmente encontrará à medida que avançamos:

- **pool** - Um pool pode ser composto por um ou vários dispositivos de bloco. O tamanho de um pool de stratis será equivalente à soma dos dispositivos de bloco que constituem o pool.
- **blockdev** - Como você deve ter adivinhado, isso se refere a dispositivos de bloco, como partições de disco.
- **Sistema de arquivos** - Um sistema de arquivos é uma camada com provisionamento fino que não tem um tamanho total fixo. O tamanho real do sistema de arquivos aumenta conforme os dados são adicionados. Stratis aumenta automaticamente o tamanho do sistema de arquivos conforme o tamanho dos dados se aproxima do tamanho virtual do sistema de arquivos.

Componentes de software do Stratis

- **Stratis-cli** - Esta é a ferramenta de linha de comando que acompanha Stratis.
- **Stratisd daemon** - Este é um daemon que cria e gerencia dispositivos de bloco e desempenha um papel no fornecimento de uma API DBUS.

Instale Stratis no RHEL

Vamos ver como você pode instalar o Stratis em seu sistema, faça o login como usuário root e execute o comando.

```
[root@localhost joatham]# yum install stratisd stratis-cli
```

Após a instalação bem-sucedida do Stratis, inicie o serviço executando o comando a seguir:

```
[root@localhost joatham]# systemctl enable stratisd --now
[root@localhost joatham]# systemctl status stratisd●
 stratisd.service - Stratis daemon
   Loaded: loaded (/usr/lib/systemd/system/stratisd.service; enabled; vendor pr>
   Active: active (running) since Wed 2021-10-06 17:54:32 -03; 30s ago
     Docs: man:stratisd(8)
```

```
  Main PID: 3691 (stratisd)
     Tasks: 1 (limit: 4811)
    Memory: 2.6M
    CGroup: /system.slice/stratisd.service
            `-3691 /usr/libexec/stratisd --log-level debug
```

Crie um pool de Stratis

Para criar um pool Stratis, você precisa de dispositivos de bloco que não estão em uso ou montados. Além disso, presume-se que o serviço Stratisd está instalado e funcionando. Por fim, os dispositivos de bloco que você usará precisam ter pelo menos `1GB` de tamanho. Para exibir os dispositivos de bloco, execute o comando `lsblk`.

```
[root@localhost joatham]# lsblk
NAME                 MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                    8:0    0   11G  0 disk
 |--sda1                 8:1    0    1G  0 part /boot
  `-sda2                 8:2    0   10G  0 part
   |--rhel-root        253:0    0  8,9G  0 lvm  /
   `-rhel-swap        253:1    0  1,1G  0 lvm  [SWAP]
sdb                    8:16   0    2G  0 disk
  `-dados-banco_de_dados
                     253:2    0    2G  0 lvm  /mnt/db
sdc                    8:32   0    2G  0 disk
 |--dados-banco_de_dados
 |                   253:2    0    2G  0 lvm  /mnt/db
 |--dados-servidor_web 253:3    0  500M  0 lvm  /mnt/www
 |--dados-swap         253:4    0    1G  0 lvm
 |--dados-sys          253:5    0  100M  0 lvm
  `-dados-extensao     253:6    0  156M  0 lvm
sdd                    8:48   0    8G  0 disk
  `-4linuxvdo          253:7    0    8G  0 vdo
    `-vg4linuxvdo-lv4linuxvdo
                     253:8    0    8G  0 lvm
sde                    8:64   0    1G  0 disk
```

Nenhum desses dispositivos de bloco deve ter uma tabela de partição. Você pode confirmar isso usando o comando.

```
[root@localhost joatham]# lsblk -p /dev/sde
NAME     MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
/dev/sde   8:64   0   1G  0 disk
```

Crie um pool Stratis a partir de um disco

Você pode criar um pool Stratis a partir de um disco único usando a sintaxe.

```
[root@localhost joatham]# stratis pool create 4linux_pool /dev/sde
[root@localhost joatham]# stratis pool list
Name                          Total Physical   Properties
4linux_pool   1 GiB / 37.63 MiB / 986.37 MiB      ~Ca,~Cr
```

Criar um sistema de arquivos a partir de um pool

Tendo criado seu pool, você pode criar um sistema de arquivos a partir de um dos pools usando a sintaxe:

```
[root@localhost joatham]# stratis filesystem create 4linux_pool 4linuxFileSystem1
[root@localhost joatham]# stratis filesystem list
Pool Name     Name                Used      Created             Device
                                             UUID
4linux_pool   4linuxFileSystem1   546 MiB   Oct 06 2021 18:04   /dev/stratis/4linux_pool/4
     linuxFileSystem1   eae60b44a102412198dda38f38a7a824
```

Montando um sistema de arquivos Stratis

Agora vamos montar os sistemas de arquivos existentes para acessá-los. Primeiro, crie os pontos de montagem.

“‘shell [root@localhost joatham]# blkid . . . /dev/sde: UUID=“76f41bb60d464e64884821b4ffdb4353” POOL_UUID=“bc355be1f76844d48286499ccc94a41e” BLOCKDEV_SECTORS=“2097152” BLOCKDEV_INITTIME=“1633554072” TYPE=“stratis” . . . /dev/mapper/stratis-1-bc355be1f76844d4828 thin-fs-eae60b44a102412198dda38f38a7a824: UUID=“eae60b44-a102-4121-98dd-a38f38a7a824” BLOCK_SIZE=“512” TYPE=“xfs”

```
[root@localhost joatham]# mkdir /4linux_stratis
```

```
[root@localhost joatham]# stratis filesystem list
Pool Name     Name                Used      Created             Device
                                             UUID
4linux_pool   4linuxFileSystem1   546 MiB   Oct 06 2021 18:04   /dev/stratis/4linux_pool/4
     linuxFileSystem1   eae60b44a102412198dda38f38a7a824
[root@localhost joatham]# mount /dev/stratis/4linux_pool/4linuxFileSystem1 /4linux_stratis/
```

Para verificar a existência dos pontos de montagem atuais, execute o comando `df`:

```
[root@localhost joatham]# df -h
Sist. Arq.

     Tam. Usado Disp. Uso% Montado em
devtmpfs

     376M     0  376M   0% /dev
tmpfs

     405M     0  405M   0% /dev/shm
tmpfs

     405M  6,4M  399M   2% /run
tmpfs

     405M     0  405M   0% /sys/fs/cgroup
/dev/mapper/rhel-root
                                                                                8,9G  6,0G
      2,9G  68% /
/dev/sda1

     1014M  323M  692M  32% /boot
/dev/mapper/dados-servidor_web
                                                                          477M  2,3M  445M

     1% /mnt/www
/dev/mapper/dados-banco_de_dados
                                                                        2,0G  6,0M  1,8G   1%

     /mnt/db
tmpfs

     81M  4,6M   77M   6% /run/user/1000
/dev/sr0

     59M   59M     0 100% /run/media/joatham/VBox_GAs_6.1.18
tmpfs

     1,0M     0  1,0M   0% /run/stratisd/keyfiles
/dev/mapper/stratis-1-bc355be1f76844d48286499ccc94a41e-thin-fs-
     eae60b44a102412198dda38f38a7a824  1,0T  7,2G 1017G   1% /4linux_stratis
```

Ou ainda o próprio comando `lsblk`:

```
[root@localhost joatham]# lsblk
NAME                                  MAJ:MIN RM  SIZE RO TYPE    MOUNTPOINT
sda                                     8:0    0   11G  0 disk
 |--sda1                                  8:1    0    1G  0 part    /boot
  `-sda2                                  8:2    0   10G  0 part
   |--rhel-root                         253:0    0  8,9G  0 lvm     /
    `-rhel-swap                         253:1    0  1,1G  0 lvm     [SWAP]
sdb                                     8:16   0    2G  0 disk
  `-dados-banco_de_dados                253:2    0    2G  0 lvm     /mnt/db
sdc                                     8:32   0    2G  0 disk
 |--dados-banco_de_dados                253:2    0    2G  0 lvm     /mnt/db
 |--dados-servidor_web                  253:3    0  500M  0 lvm     /mnt/www
```

```
  |--dados-swap                                253:4    0    1G  0 lvm
  |--dados-sys                                 253:5    0  100M  0 lvm
  `-dados-extensao                             253:6    0  156M  0 lvm
sdd                                            8:48   0    8G  0 disk
  `-4linuxvdo                                  253:7    0    8G  0 vdo
    `-vg4linuxvdo-lv4linuxvdo                  253:8    0    8G  0 lvm
sde                                            8:64   0    1G  0 disk
  `-stratis-1-private-bc355be1f76844d48286499ccc94a41e-physical-originsub
                                             253:9    0 1020M  0 stratis
   |--stratis-1-private-bc355be1f76844d48286499ccc94a41e-flex-thinmeta
   |                                         253:10   0   16M  0 stratis
   |   `-stratis-1-private-bc355be1f76844d48286499ccc94a41e-thinpool-pool
   |                                         253:13   0  972M  0 stratis
   |     `-stratis-1-bc355be1f76844d48286499ccc94a41e-thin-fs-
         eae60b44a102412198dda38f38a7a824
   |                                         253:14   0    1T  0 stratis /4linux_stratis
   |--stratis-1-private-bc355be1f76844d48286499ccc94a41e-flex-thindata
   |                                         253:11   0  972M  0 stratis
   |   `-stratis-1-private-bc355be1f76844d48286499ccc94a41e-thinpool-pool
   |                                         253:13   0  972M  0 stratis
   |     `-stratis-1-bc355be1f76844d48286499ccc94a41e-thin-fs-
         eae60b44a102412198dda38f38a7a824
   |                                         253:14   0    1T  0 stratis /4linux_stratis
    `-stratis-1-private-bc355be1f76844d48286499ccc94a41e-flex-mdv
                                             253:12   0   16M  0 stratis
```

Outras possibilidades de conferências são:

```
[root@localhost joatham]# stratis blockdev
Pool Name     Device Node   Physical Size   Tier
4linux_pool   /dev/sde              1 GiB   Data
```

Sistemas de arquivos Stratis de montagem persistente

Os pontos de montagem que acabamos de criar não sobrevivem a uma reinicialização. Para torná-los persistentes, primeiro obtenha o `UUID` de cada um dos sistemas de arquivos:

```
[root@localhost joatham]# stratis fs
Pool Name     Name               Used      Created             Device
                                           UUID
4linux_pool   4linuxFileSystem1   546 MiB   Oct 06 2021 18:04   /dev/stratis/4linux_pool/4
     linuxFileSystem1   eae60b44a102412198dda38f38a7a824
```

```
[root@localhost joatham]# echo "/dev/stratis/4linux_pool/4linuxFileSystem1      /4
     linux_stratis xfs     defaults        0       0
" | tee-a /etc/fstab
```

```
[root@localhost joatham]# init 6
```

Criando Snapshots com Stratis

Um *Snapshots* é uma leitura com provisionamento *thin* e grava uma cópia de um sistema de arquivos em um determinado momento.

Para criar um Snapshots, execute os seguintes passos:

- 1 - Precisamos criar algum conteudo dentro do pool, portanto:

```
[root@localhost joatham]# echo "4Linux.. formando so os melhores" > /4linux_stratis/
     ArquivoTeste.txt
[root@localhost joatham]# cat /4linux_stratis/ArquivoTeste.txt
4Linux.. formando so os melhores
```

- 2 - Verifique as informações do seu FileSystem Stratis

```
[root@localhost joatham]# stratis fs
Pool Name     Name               Used      Created             Device
                                          UUID
4linux_pool   4linuxFileSystem1  546 MiB   Oct 06 2021 18:04   /dev/stratis/4linux_pool/4
     linuxFileSystem1   eae60b44a102412198dda38f38a7a824
```

- 3 - Execute o comando de *Snapshot* com os parametros de `Pool` e `Name`, respectivamente, como mostra o comando abaixo:

```
[root@localhost joatham]# stratis filesystem snapshot 4linux_pool 4linuxFileSystem1
     Snapshot1
```

- 4 - Observe que o Snapshot foi criado com sucesso:

```
[root@localhost joatham]# stratis fs
Pool Name     Name               Used      Created             Device
                                          UUID
4linux_pool   4linuxFileSystem1  546 MiB   Oct 06 2021 18:04   /dev/stratis/4linux_pool/4
     linuxFileSystem1   eae60b44a102412198dda38f38a7a824
4linux_pool   Snapshot1          546 MiB   Oct 06 2021 21:09   /dev/stratis/4linux_pool/
     Snapshot1          29f8533873d24220b415ecc3e4ee6b7d
```

- 5 - Agora, remova o conteúdo criado e desmonte o *pool*:

```
[root@localhost joatham]# rm /4linux_stratis/ArquivoTeste.txt
rm: remover arquivo comum '/4linux_stratis/ArquivoTeste.txt'? y
[root@localhost joatham]# umount /4linux_stratis
```

- 6 - Ajuste os pontos de montagem para atender o *Snapshot*

```
[root@localhost joatham]# stratis fs list
Pool Name     Name                Used      Created             Device
                                             UUID
4linux_pool   4linuxFileSystem1   546 MiB   Oct 06 2021 18:04   /dev/stratis/4linux_pool/4
     linuxFileSystem1   eae60b44a102412198dda38f38a7a824
4linux_pool   Snapshot1           546 MiB   Oct 06 2021 21:09   /dev/stratis/4linux_pool/
     Snapshot1           29f8533873d24220b415ecc3e4ee6b7d
```

```
[root@localhost joatham]# echo "/dev/stratis/4linux_pool/Snapshot1      /4linux_stratis xfs
          defaults        0       0
" | tee -a /etc/fstab
```

> Nao se esqueça de apagar a linha do ponto de montagem anterior do `fstab`

Removendo um sistema de arquivos Stratis

Para remover um sistema de arquivos, você precisa, em primeiro lugar, desmontar o sistema de arquivos conforme mostrado.

```
[root@localhost joatham]# umount /4linux_stratis

[root@localhost joatham]# stratis fs destroy 4linux_pool Snapshot1

[root@localhost joatham]# stratis fs
Pool Name     Name                Used      Created             Device
                                             UUID
4linux_pool   4linuxFileSystem1   546 MiB   Oct 06 2021 18:04   /dev/stratis/4linux_pool/4
     linuxFileSystem1   eae60b44a102412198dda38f38a7a824
```

Adicionando um disco a um pool de Stratis existente

Você pode adicionar um disco a um pool existente usando o comando:

```
[root@localhost joatham]# stratis pool add-data 4linux_pool /dev/sdX
```

> Dica: para remover e listar um pool Stratis, os comandos são:

```
stratis pool destroy my_pool
```

```
stratis pool list
```

19

Diagnosticar e corrigir problemas de permissão de arquivo

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Criar e configurar sistemas de arquivos**
  - Crie, monte, desmonte e use sistemas de arquivos vfat, ext4 e xfs
  - Monte e desmonte sistemas de arquivos de rede usando NFS
  - Amplie os volumes lógicos existentes
  - Crie e configure diretórios set-GID para colaboração
  - Configure compactação de disco
  - Gerencie armazenamento em camadas
  - **Diagnostique e corrija problemas de permissão de arquivo**

Hands On

- Criação de dois usuários com suas respectivas senhas:

```
[root@localhost joatham]# adduser usuario1
[root@localhost joatham]# passwd usuario1
```

```
Mudando senha para o usuário usuario1.
Nova senha:
SENHA INCORRETA: A senha é menor do que 8 caracteres
Redigite a nova senha:
passwd: todos os tokens de autenticações foram atualizados com sucesso.
```

- Criação de dois grupos:

```
[root@localhost joatham]# groupadd TI
[root@localhost joatham]# groupadd RH
```

- Adicionar os usuários aos seus respectivos grupos:

```
[root@localhost joatham]# gpasswd -a usuario1 TI
Adicionando usuário usuario1 ao grupo TI
[root@localhost joatham]# gpasswd -a usuario2 RH
Adicionando usuário usuario2 ao grupo RH
```

- Vamos criar uma estrutura de pastas para dar mais realidade ao exemplo.

```
[root@localhost joatham]# mkdir /4linux
[root@localhost joatham]# cd /4linux/
[root@localhost 4linux]# mkdir TI
[root@localhost 4linux]# mkdir RH
[root@localhost 4linux]#
[root@localhost 4linux]# cd TI/
[root@localhost TI]# mkdir pasta_publica
[root@localhost TI]# mkdir pasta_privada
```

```
[root@localhost /]# tree 4linux/
4linux/
 |--- RH
  `-- TI
     |--- pasta_privada
      `-- pasta_publica

4 directories, 0 files
```

- Ajustando as permissões:

```
[root@localhost 4linux]# chmod -R 770 TI/
[root@localhost 4linux]# chmod -R 770 RH/
```

- Colocando o usuário dentro do grupo:

```
[joatham@localhost ~]$ newgrp TI
Senha :
newgrp: failed to crypt password with previous salt: Argumento inválido
```

```
[root@localhost joatham]# usermod -aG TI joatham
```

```
[joatham@localhost ~]$ ls /4linux/TI/
ls: não foi possível abrir o diretório '/4linux/TI/': Permissão negada
```

```
[joatham@localhost ~]$ su -l joatham
Senha:
[joatham@localhost ~]$ ls /4linux/TI/
pasta_privada  pasta_publica
```

- Ajustando permissões em arquivos:

```
[joatham@localhost pasta_privada]$ touch joatham.txt
[joatham@localhost pasta_privada]$ ls -l
total 0
-rw-rw-r--. 1 joatham joatham 0 nov  3 16:15 joatham.txt
```

```
[usuario1@localhost pasta_privada]$ rm -rf joatham.txt
[usuario1@localhost pasta_privada]$ ls -l
total 0
[usuario1@localhost pasta_privada]$
```

- Aplicando permissão Srick Bit na pasta:

```
[root@localhost TI]# chmod -R 1770 pasta_privada/
[root@localhost TI]# ls -l
total 0
drwxrwx--T. 2 root TI 6 nov  3 16:17 pasta_privada
drwxrwx---. 2 root TI 6 nov  3 15:24 pasta_publica
[root@localhost TI]#
```

```
[usuario1@localhost pasta_privada]$ rm -rf joatham.txt
rm: não foi possível remover 'joatham.txt': Operação não permitida
[usuario1@localhost pasta_privada]$
```

20

Agendar tarefas usando at e cron

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Implantar, configurar e manter sistemas**
  - **Agende tarefas usando at e cron**
  - Inicie e pare serviços, além de configurar serviços para iniciar automaticamente durante a inicialização
  - Configure os sistemas para inicializar em um destino específico automaticamente
  - Configure clientes de serviço de tempo
  - Instale e atualize pacotes de software da Red Hat Network, um repositório remoto ou do sistema de arquivos local
  - Trabalhe com fluxos de módulo de pacote
  - Modifique o bootloader do sistema

Crontab

`Cron` é um daemon de agendamento que executa tarefas em intervalos especificados. Essas tarefas são chamadas de *cron jobs* e são usadas principalmente para automatizar a manutenção ou administração do sistema.

Por exemplo, você pode definir um *cron job* para automatizar tarefas repetitivas, como backup

de bancos de dados, atualização do sistema com os patches de segurança mais recentes, verificação do uso de espaço em disco, envio de e-mails e assim por diante.

Os *cron jobs* podem ser programados para execução por minuto, hora, dia do mês, mês, dia da semana ou qualquer combinação similar.

O que é o arquivo `Crontab`

`Crontab` (tabela cron) é um arquivo de texto que especifica a programação de tarefas `cron` . Existem dois tipos de arquivos `crontab`: os arquivos `crontab` de todo o sistema e arquivos `crontab` de usuários individuais.

Os arquivos `crontab` dos usuários são nomeados de acordo com o nome do usuário. Sua localização varia de acordo com os sistemas operacionais. Em distribuições baseadas em Red Hat, como CentOS, os arquivos `crontab` são armazenados no diretório `/var/spool/cron`, enquanto no Debian e no Ubuntu os arquivos são armazenados no diretório `/var/spool/cron/crontabs`.

Embora você possa editar os arquivos `crontab` do usuário manualmente, é recomendável usar o comando `crontab`.

O arquivo `/etc/crontab` e os scripts dentro do diretório `/etc/cron.d` são arquivos `crontab` de todo o sistema que podem ser editados apenas pelos administradores do sistema.

Na maioria das distribuições Linux, você também pode colocar scripts dentro dos diretórios `/etc/cron.{hourly,daily,weekly,monthly}`, e os scripts serão executados a cada `hour/day/week/month`.

Sintaxe e operadores do `Crontab`

Cada linha no arquivo `crontab` do usuário contém seis campos, separados por um espaço e seguido pelo comando a ser executado.

```
* * * * * command(s)
- - - - -
| | | | |
| | | | ----- Day of week (0 - 7) (Sunday=0 or 7)
| | | ------- Month (1 - 12)
| | --------- Day of month (1 - 31)
| ----------- Hour (0 - 23)
------------- Minute (0 - 59)
```

Os primeiros cinco campos podem conter um ou mais valores separados por vírgula ou um

intervalo de valores separados por hífen.

- `*` - O operador asterisco significa qualquer valor ou sempre. Se você tiver o símbolo de asterisco no campo Hora, significa que a tarefa será realizada a cada hora.
- `,` - O operador vírgula permite que você especifique uma lista de valores para repetição. Por exemplo, se você tiver 1,3,5 no campo Hora, a tarefa será executada às 1h, 3h e 5h.
- `-` - O operador hífen permite especificar um intervalo de valores. Se você tiver 1-5 no campo Dia da semana, a tarefa será executada todos os dias da semana (de segunda a sexta-feira).
- `/` - O operador de barra permite especificar valores que serão repetidos em um determinado intervalo entre eles. Por exemplo, se você tiver `*/4` no campo Hora, significa que a ação será realizada a cada quatro horas. É o mesmo que especificar 0,4,8,12,16,20. Em vez do asterisco antes do operador de barra, você também pode usar um intervalo de valores, 1-30/10 significa o mesmo que 1,11,21.

Macros Predefinidas

Existem várias macros de cronograma `Cron` especiais usadas para especificar intervalos comuns. Você pode usar esses atalhos no lugar da especificação de data de cinco colunas.

- **`@yearly`**(ou @annually) - Executar a tarefa especificada uma vez por ano à meia-noite (12h00) de 1ž de janeiro. Equivalente a 0 0 1 1 *.
- **`@monthly`**- Execute a tarefa especificada uma vez por mês à meia-noite do primeiro dia do mês. Equivalente a 0 0 1 * *.
- **`@weekly`**- Execute a tarefa especificada uma vez por semana à meia-noite de domingo. Equivalente a 0 0 * * 0.
- **`@daily`**- Execute a tarefa especificada uma vez por dia à meia-noite. Equivalente a 0 0 * * *.
- **`@hourly`**- Execute a tarefa especificada uma vez por hora no início da hora. Equivalente a 0 * * * *.
- **`@reboot`** - Execute a tarefa especificada na inicialização do sistema (tempo de inicialização).

Comando Linux Crontab

O comando `crontab` permite que você instale, visualize ou abra um arquivo `crontab` para edição: - `crontab -e` - Edite o arquivo crontab ou crie um se ainda não existir. - `crontab -l` - Exibir o conteúdo do arquivo crontab. - `crontab -r` - Remova seu arquivo crontab atual. - `crontab -i` - Remova seu arquivo crontab atual com um prompt antes da remoção. - `crontab -u` - Edite outro arquivo crontab do usuário. Esta opção requer privilégios de administrador do sistema.

O comando `crontab` abre o arquivo `crontab` usando o editor especificado pelas variáveis de ambiente `VISUAL` ou `EDITOR`.

Restrições `Crontab`

Os arquivos `/etc/cron.deny` e `/etc/cron.allow` permitem controlar quais usuários têm acesso ao comando `crontab`. Os arquivos consistem em uma lista de nomes de usuário, um nome de usuário por linha.

Por padrão, apenas o `/etc/cron.deny` arquivo existe e está vazio, o que significa que todos os usuários podem usar o comando `crontab`. Se você deseja negar acesso aos comandos `crontab` para um usuário específico, adicione o nome de usuário a este arquivo.

Se o arquivo `/etc/cron.allow` existir, apenas os usuários listados neste arquivo podem usar o comando `crontab`.

Se nenhum dos arquivos existir, apenas os usuários com privilégios administrativos podem usar o comando `crontab`

Exemplos de Cron Jobs

Abaixo estão alguns exemplos de *cron job* que mostram como agendar uma tarefa para ser executada em diferentes períodos de tempo.

- Execute um comando às 15:00 todos os dias, de segunda a sexta-feira:

```
0 15 * * 1-5 commando
```

- Execute um script a cada 5 minutos e redirecione a saída padrão para dev null, apenas o erro padrão será enviado para o endereço de e-mail especificado:

```
MAILTO=email@example.com
*/5 * * * * /path/to/script.sh > /dev/null
```

- Execute dois comandos todas as segundas-feiras às 15h (use o operador `&&` entre os comandos):

```
0 15 * * Mon commando1 && commando2
```

- Execute um script PHP a cada 2 minutos e grave a saída em um arquivo:

```
*/2 * * * * /usr/bin/php /path/to/script.php >> /var/log/script.log
```

- Execute um script todos os dias, a cada hora, das 8h às 16h:

```
00 08-16 * * * /path/to/script.sh
```

- Execute um script na primeira segunda-feira de cada mês, às 7h:

```
0 7 1-7 * 1 /path/to/script.sh
```

Execute o script às 21h15, nos dias 1ž e 15 de cada mês:

```
15 9 1,15 * * /path/to/script.sh
```

Agendando tarefas com o comando `at` no Linux

O comando `at` no Linux é uma excelente forma de agendar tarefas/comandos que devem ser executadas apenas uma vez no Linux. É importante ter em mente que seu uso é indicado para tarefas únicas, aquelas que devem ser executadas em uma data específica. Para tarefas recorrentes o cron e crontab são mais adequados.

Uma lista dos comandos que devem ser conhecidos para um bom gerenciamento de tarefas:

- o comando `at` executa comandos em um data/hora específicas;
- o comando `atq` lista as tarefas agendadas de um determinado usuário, já para um superusuário o comando `atq` irá listar as tarefas de todos os usuários;
- o comando `atrm` é utilizado para deletar as tarefas agendadas pelo `at`.

O comando `at`

A sintaxe básica do comando at é bem simples (at [opções] HORA). Algumas opções úteis para o comando at:

- -m - Envia um email para o usuário quando a tarefa for completada
- -M - Nunca envia o email para usuário
- -f - Lê as tarefas do arquivo especificado
- -c - Exibe as tarefas agendas na tela
- -t - Especifica a data/hora da tarefa, no formato YYYYMMDDhhmm# Inicializar e interromper serviços, bem como configurá-los para iniciar automaticamente na inicialização ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:
- **Implantar, configurar e manter sistemas**
  - Agende tarefas usando at e cron
  - **Inicie e pare serviços, além de configurar serviços para iniciar automaticamente durante a inicialização**
  - Configure os sistemas para inicializar em um destino específico automaticamente
  - Configure clientes de serviço de tempo
  - Instale e atualize pacotes de software da Red Hat Network, um repositório remoto ou do sistema de arquivos local
  - Trabalhe com fluxos de módulo de pacote
  - Modifique o bootloader do sistema

Systemctl

No Linux, um serviço é um programa executado em segundo plano. Os serviços podem ser iniciados sob demanda ou no momento da inicialização.

Se você estiver usando Linux como seu sistema operacional primário ou plataforma de desenvolvimento, você lidará com diferentes serviços como `servidor web`, `ssh` ou `cron`.

Saber como listar os serviços em execução ou verificar o status do serviço é importante ao depurar problemas do sistema.

A maioria das distribuições recentes do Linux está usando o `systemd` como sistema `init` padrão e gerenciador de serviço.

`Systemd` é um conjunto de ferramentas para gerenciamento de sistemas Linux. É usado para inicializar a máquina, gerenciar serviços, montar sistemas de arquivos automaticamente, registrar eventos, configurar o nome do host e outras tarefas do sistema. Ou seja, exatamente o que buscamos.

Listagem de serviços Linux

O `Systemd` usa o conceito de unidades, que podem ser serviços, soquetes, pontos de montagem, dispositivos, etc. As unidades são definidas usando arquivos de texto em iniformato. Esses arquivos incluem informações sobre a unidade, suas configurações e comandos a serem executados. As extensões de nome de arquivo definem o tipo de arquivo da unidade. Por exemplo, os arquivos da unidade de serviço do sistema têm uma extensão `.service`.

`systemctl` é um utilitário de linha de comando usado para controlar o `systemd` e gerenciar serviços. Faz parte do ecossistema `systemd` e está disponível por padrão em todos os sistemas. Para obter uma lista de todas as unidades de serviço carregadas, digite:

```
[root@localhost joatham]# systemctl list-unit --type service
Unknown operation list-unit.
[root@localhost joatham]# systemctl list-units --type service
  UNIT                                LOAD   ACTIVE SUB     DESCRIPTION
  accounts-daemon.service             loaded active running Accounts Service
  alsa-state.service                  loaded active running Manage Sound Card State (restore
        and store)
  atd.service                         loaded active running Job spooling tools
  auditd.service                      loaded active running Security Auditing Service
  avahi-daemon.service                loaded active running Avahi mDNS/DNS-SD Stack
  chronyd.service                     loaded active running NTP client/server
  colord.service                      loaded active running Manage, Install and Generate
       Color Profiles
  crond.service                       loaded active running Command Scheduler
  cups.service                        loaded active running CUPS Scheduler
  dbus.service                        loaded active running D-Bus System Message Bus
  dm-event.service                    loaded active running Device-mapper event daemon●
 dnf-makecache.service               loaded failed failed  dnf makecache
```

Cada linha de saída contém as seguintes colunas da esquerda para a direita:

- **UNIT** - O nome da unidade de serviço.
- **LOAD** - Informações sobre se o arquivo da unidade foi carregado na memória.
- **ACTIVE** - O estado de ativação do arquivo de unidade de alto nível, que pode ser ativo, recarregando, inativo, com falha, ativando, desativando. É uma generalização da SUBcoluna.
- **SUB** - O estado de ativação do arquivo da unidade de baixo nível. O valor deste campo depende do tipo de unidade. Por exemplo, uma unidade do tipo serviço pode estar em um dos seguintes estados: inativo, encerrado, com falha ou em execução.
- **DESCRIPTION** - Breve descrição do arquivo da unidade.

Por padrão, o comando lista apenas as unidades ativas carregadas. Para ver as unidades carregadas, mas também inativas, passe a opção `--all`:

```
[root@localhost joatham]# systemctl list-units --type service --all
```

Se você quiser ver todos os arquivos da unidade instalados, não apenas os carregados, use:

```
[root@localhost joatham]# systemctl list-unit-files
```

Controlar e gerenciar serviços usando `Systemctl`

- Liste todos os serviços (incluindo habilitados e desabilitados).

```
[root@localhost joatham]# systemctl list-unit-files --type service
UNIT FILE                                  STATE
accounts-daemon.service                    enabled
alsa-restore.service                       static
alsa-state.service                         static
anaconda-direct.service                    static
anaconda-fips.service                      static
anaconda-nm-config.service                 static
anaconda-noshell.service                   static
anaconda-pre.service                       static
anaconda-shell@.service                    static
anaconda-sshd.service                      static
anaconda-tmux@.service                     static
anaconda.service                           static
arp-ethers.service                         disabled
atd.service                                enabled
```

- Como eu inicio, reinicio, paro, recarrego e verifico o status de um serviço (**crond.service**) no Linux:

```
systemctl start crond.service
[root@localhost joatham]# systemctl restart crond.service
[root@localhost joatham]# systemctl stop crond.service
[root@localhost joatham]# systemctl reload crond.service
[root@localhost joatham]# systemctl status crond.service
```

```
[root@localhost joatham]# systemctl status crond.service●
 crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-10-06 21:33:35 -03; 22h ago
 Main PID: 1310 (crond)
    Tasks: 1 (limit: 4811)
   Memory: 324.0K
```

```
   CGroup: /system.slice/crond.service
           `-1310 /usr/sbin/crond -n
```

> Quando usamos comandos como `start`, `restart`, `stop` e `reload` com `systemctl`, não obteremos nenhuma saída no terminal. O único comando que imprimirá a saída é `status`.

- Como ativar um serviço e habilitar ou desabilitar um serviço no momento da inicialização (serviço de inicialização automática na inicialização do sistema).

```
[root@localhost joatham]# systemctl is-active crond.service
active
```

```
[root@localhost joatham]# systemctl disable crond.service
Removed /etc/systemd/system/multi-user.target.wants/crond.service.
```

```
[root@localhost joatham]# systemctl enable crond.service
Created symlink /etc/systemd/system/multi-user.target.wants/crond.service → /usr/lib/
     systemd/system/crond.service.
```

- Como mascarar (tornando impossível iniciar) ou desmascarar um serviço (crond.service).

```
[root@localhost joatham]# systemctl mask crond.service
Created symlink /etc/systemd/system/crond.service → /dev/null.
```

```
[root@localhost joatham]# systemctl unmask crond.service
Removed /etc/systemd/system/crond.service.
```

- Como matar um serviço usando o comando `systemctl`.

`shell [root@localhost joatham]# systemctl kill httpd [root@localhost joatham]# systemctl status httpd`# Configurar sistemas para inicializar em um determinado destino automaticamente.

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Implantar, configurar e manter sistemas**
  - Agende tarefas usando at e cron
  - Inicie e pare serviços, além de configurar serviços para iniciar automaticamente durante a inicialização
  - **Configure os sistemas para inicializar em um destino específico automaticamente**
  - Configure clientes de serviço de tempo
  - Instale e atualize pacotes de software da Red Hat Network, um repositório remoto ou do sistema de arquivos local
  - Trabalhe com fluxos de módulo de pacote
  - Modifique o bootloader do sistema

Hands On

- Vizualizando os vários `targets` do seu sistema:

```
[root@localhost joatham]# systemctl list-units --type=target
UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
cryptsetup.target      loaded active active Local Encrypted Volumes
getty.target           loaded active active Login Prompts
graphical.target       loaded active active Graphical Interface
local-fs-pre.target    loaded active active Local File Systems (Pre)
local-fs.target        loaded active active Local File Systems
multi-user.target      loaded active active Multi-User System
network-online.target  loaded active active Network is Online
network-pre.target     loaded active active Network (Pre)
network.target         loaded active active Network
nfs-client.target      loaded active active NFS client services
nss-user-lookup.target loaded active active User and Group Name Lookups
paths.target           loaded active active Paths
remote-fs-pre.target   loaded active active Remote File Systems (Pre)
remote-fs.target       loaded active active Remote File Systems
rpc_pipefs.target      loaded active active rpc_pipefs.target
rpcbind.target         loaded active active RPC Port Mapper
slices.target          loaded active active Slices
sockets.target         loaded active active Sockets
sound.target           loaded active active Sound Card
sshd-keygen.target     loaded active active sshd-keygen.target
swap.target            loaded active active Swap
sysinit.target         loaded active active System Initialization
timers.target          loaded active active Timers

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.
```

```
24 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

- Vizualizando o alvo padrão de inicialização do sistema:

```
[root@localhost joatham]# systemctl get-default
graphical.target
```

- Alterar o alvo de incialização:

```
[root@localhost joatham]# systemctl set-default multi-user.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/multi-user.
     target.
```

```
[root@localhost joatham]# systemctl get-default
multi-user.target
```

```
[root@localhost joatham]# init 6
```

> Agora, dê uma olhada na sua máquina virtual!!

- Volte à configuração gráfica do seu sistema.

```
[root@localhost joatham]# systemctl set-default grafical-user
```

```
[root@localhost joatham]# init 6
```

> Novamente, dê uma olhada na sua maquina virtual!! # Configurar clientes de serviço de tempo ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Implantar, configurar e manter sistemas**
  - Agende tarefas usando at e cron
  - Inicie e pare serviços, além de configurar serviços para iniciar automaticamente durante a inicialização
  - Configure os sistemas para inicializar em um destino específico automaticamente
  - **Configure clientes de serviço de tempo**
  - Instale e atualize pacotes de software da Red Hat Network, um repositório remoto ou do sistema de arquivos local
  - Trabalhe com fluxos de módulo de pacote
  - Modifique o bootloader do sistema

Introdução

Embora os computadores pessoais sejam capazes de manter a hora razoavelmente precisa por conta própria, a computação de produção e os ambientes de rede exigem que um tempo muito preciso seja mantido. O tempo mais preciso é medido por relógios de referência, que normalmente são relógios atômicos. O mundo moderno desenvolveu um programa em que todos os sistemas de computador conectados à Internet podem ser sincronizados com esses relógios de referência, usando o que é conhecido como *Network Time Protocol* (NTP). Um sistema de computador com NTP será capaz de sincronizar os relógios do sistema com a hora fornecida pelos relógios de referência. Se a hora do sistema e a hora medida nesses servidores forem diferentes, o computador aumentará ou diminuirá a hora do sistema interno de forma incremental até que a hora do sistema corresponda à hora da rede.

NTP é o protocolo que garante a sincronização horária numa rede, onde um ou mais servidores mestre vão facultando as horas ás maquinas clientes. Note que isto é baseado num sistema hierárquico, sendo que os próprios clientes podem ser outros servidores que estão a facultar as horas a outras máquinas e assim sucessivamente. Num sistema RHEL existem 2 pacotes, principalmente usados para gerir o protocolo NTP, sendo esses os seguintes:

- NTP - Este é o pacote clássico que foi principalmente usado em instalações do RHEL6 e RHEL5. Este pacote pode ser usado como cliente ou servidor.
- Chrony - Esta é a nova solução desenvolvida e mais capaz de lidar com máquinas portáteis e máquinas com limitações de banda de larga, pois este pacote é capaz de garantir a sincronização horária de forma mais rápida. Note que este pacote é majoritariamente usado para máquinas cliente.

NTP

NTP Daemon O horário do sistema é comparado ao horário da rede em uma programação regular. Para que isso funcione, devemos ter um daemon rodando em segundo plano. Para

muitos sistemas Linux, o nome desse daemon é ntpd. ntpdpermitirá que uma máquina não seja apenas uma consumidora de tempo (ou seja, capaz de sincronizar seu próprio relógio de uma fonte externa), mas também forneça tempo para outras máquinas.

Vamos supor que nosso computador seja baseado em systemd e use systemctlpara controlar daemons. Vamos instalar os ntppacotes usando o gerenciador apropriado e, em seguida, garantir que nosso ntpddaemon esteja em execução, verificando seu status:

```
[root@localhost joatham]# systemctl status ntpd●

 ntpd.service - Serviço de tempo de rede
   Carregado: carregado (/usr/lib/systemd/system/ntpd.service; ativado; predefinição do
        fornecedor: desativado)
   Ativo: ativo (em execução) desde sexta-feira 06-12-2019 03:27:21 EST; 7h atrás
  Processo: 856 ExecStart = / usr / sbin / ntpd -u ntp: ntp $ OPTIONS (código = encerrado,
       status = 0 / SUCESSO)
 PID principal: 867 (ntpd)
   CGroup: /system.slice/ntpd.service
           `-867 / usr / sbin / ntpd -u ntp: ntp -g
```

Em alguns casos, pode ser necessário iniciar e ativar ntpd. Na maioria das máquinas Linux, isso é feito com:

```
[root@localhost joatham]# systemctl enable ntpd && systemctl start ntpd
```

As consultas NTP acontecem na porta 123TCP.

O serviço é capaz de pesquisar várias fontes e selecionar os melhores candidatos a serem usados para definir a hora do sistema. Se uma conexão de rede for perdida, o NTP usa ajustes anteriores de seu histórico para estimar ajustes futuros.

Dependendo da sua distribuição do Linux, a lista de servidores de horário da rede será armazenada em locais diferentes. Vamos supor que ntp está instalado em sua máquina.

O arquivo /etc/ntp.conf contém informações de configuração sobre como o sistema sincroniza com o horário da rede. Este arquivo pode ser lido e modificado usando vi ou nano.

Por padrão, os servidores NTP usados serão especificados em uma seção como esta:

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.centos.pool.ntp.org iburst
```

```
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
```

A sintaxe para adicionar servidores NTP é assim:

```
server (IP Address)
server server.url.localhost
```

Os endereços do servidor podem ser endereços IP ou URLs se o DNS tiver sido configurado corretamente. Nesse caso, o servidor sempre será consultado.

Um administrador de rede também pode considerar o uso (ou configuração) de um pool. Neste caso, assumimos que existem vários provedores `NTP`, todos executando daemons NTP e com o mesmo tempo. Quando um cliente consulta um pool, um provedor é selecionado aleatoriamente. Isso ajuda a distribuir a carga da rede entre muitas máquinas, de forma que nenhuma máquina no pool esteja lidando com todas as consultas NTP.

Normalmente, `/etc/ntp.conf` será preenchido com um pool de servidores chamado `pool.ntp.org`. Portanto, por exemplo, `server0.centos.pool.ntp.org` é um pool `NTP` padrão fornecido para máquinas CentOS.

- **`ntpdate`** - Durante a configuração inicial, a hora do sistema e o `NTP` podem ser seriamente dessincronizados. Se o deslocamento entre o sistema e a hora `NTP` for maior que 17 minutos, o daemon `NTP` não fará alterações na hora do sistema. Neste cenário, a intervenção manual será necessária.

Em primeiro lugar, se o daemon `ntpd` estiver em execução será necessário interromper o serviço. Use `systemctl stop ntpd` para fazer isso.

Em seguida, use `ntpdate pool.ntp.org` para executar uma sincronização inicial única, em que `pool.ntp.org` se refere ao endereço IP ou URL de um servidor NTP.

- **`ntpq`** - `ntpq` é um utilitário para monitorar o status do NTP. Uma vez que o daemon NTP foi iniciado e configurado, `ntpq` pode ser usado para verificar seu status:

```
[root@localhost joatham]# ntpq -p
     remoto refid st t quando poll alcance atraso offset offset jitter
==============================================================================
+37,44,185,42 91,189,94,4 3 u 86 128 377 126,509 -20,398 6,838
+ ntp2.0x00.lv 193.204.114.233 2 u 82 128 377 143,885 -8,105 8,478
* inspektor-vlan1 121.131.112.137 2 u 17 128 377 112,878 -23,619 7,959
```

```
b1-66er.matrix. 18.26.4.105 2 u 484 128 10 34.907 -0.811 16.123
```

chrony

chrony é mais uma maneira de implementar o NTP. Ele é instalado por padrão em alguns sistemas Linux, mas está disponível para download em todas as principais distribuições. chronyd é o daemon, enquanto chrony e chronyc é a interface da linha de comandos. Pode ser necessário iniciar e ativar chronyd antes de interagir com chronyc.

Hands on

- Checagem de repositórios do RHEL

```
[root@localhost joatham]# dnf repolist
Updating Subscription Management repositories.
id do repo                      nome do repo
rhel-8-for-x86_64-appstream-rpms Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
rhel-8-for-x86_64-baseos-rpms    Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)
```

- Instalação do pacote chrony

```
[root@localhost joatham]# dnf install chrony* -y
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 0:11:41 atrás em sáb 09 out 2021
     15:39:44 -03.
O pacote chrony-3.5-2.el8.x86_64 já está instalado.
Dependências resolvidas.
Nada para fazer.
Concluído!
```

```
[root@localhost joatham]# systemctl status chronyd●
 chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled;>
   Active: active (running) since Sat 2021-10-09 14:45:50 -03; 1h 7m>
     Docs: man:chronyd(8)
           man:chrony.conf(5)
  Process: 1043 ExecStartPost=/usr/libexec/chrony-helper update-daem>
  Process: 1031 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, s>
 Main PID: 1041 (chronyd)
    Tasks: 1 (limit: 19972)
   Memory: 2.4M
   CGroup: /system.slice/chronyd.service
           `-1041 /usr/sbin/chronyd
```

- Uma maneira interessante de verificar as origens do chrony é executando o comando abaixo:

```
[root@localhost joatham]# chronyc sources -v
210 Number of sources = 5

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
| /   '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.           |  xxxx = adjusted offset,
||      Log2(Polling interval) --.      |          |  yyyy = measured offset,
||                                \     |          |  zzzz = estimated error.
||                                 |    |           \
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* a.ntp.br                      2   6   377     3   +541us[ +750us] +/-   16ms
^+ 45.11.105.243                 2   6   375     3   +719us[ +719us] +/-   16ms
^- time100.stupi.se              1   6   377    65  -2417us[-2360us] +/-  117ms
^- time.cloudflare.com           3   6   377     1  +9428us[+9428us] +/-   78ms
^+ stratum2-1.ntp.sao03.br.>     2   6   377     3   +651us[ +651us] +/-   16ms
```

- Editando o arquivo de configuração padrão do serviço `chrony`.

```
[root@localhost joatham]# vi +23 /etc/chrony.conf
...
...
# Allow NTP client access from local network.
allow 10.11.10.0/16
```

```
[root@localhost joatham]# systemctl restart chronyd.service
```

```
[root@localhost joatham]# systemctl status chronyd.service●
 chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled;>
   Active: active (running) since Sat 2021-10-09 15:59:08 -03; 16s a>
     Docs: man:chronyd(8)
           man:chrony.conf(5)
  Process: 4453 ExecStartPost=/usr/libexec/chrony-helper update-daem>
  Process: 4449 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, s>
 Main PID: 4451 (chronyd)
    Tasks: 1 (limit: 19972)
   Memory: 512.0K
   CGroup: /system.slice/chronyd.service
           `-4451 /usr/sbin/chronyd
```

- Adicione o serviço `NTP` ao seu `firewall` para que outros clientes possam se conectar a

você.

```
[root@localhost joatham]# firewall-cmd --permanent --add-service=ntp
```

- Recarregue seu firewall para que as configurações possam fazer efeito.

```
[root@localhost joatham]# firewall-cmd --reload
```

- Cheque quais clientes estão ativos no servidor `NTP` via `chronyd`

```
[root@localhost joatham]# chronyc clients
Hostname                      NTP   Drop Int IntL Last     Cmd   Drop Int  Last
===============================================================================
[root@localhost joatham]#
```

Pronto, Servidor configurado! Agora vamos ajustar nosso cliente:

```
root@kali:~# aptitude search chrony
p   chrony

     - Versatile implementation of the Network Time Protocol
p   puppet-module-aboe-chrony                                                                                    -

     Puppet module for Chrony
p   ruby-em-synchrony

     - fiber aware EventMachine libraries
```

```
root@kali:~# aptitude install chrony
```

```
root@kali:~# /etc/init.d/chrony start
Starting chrony (via systemctl): chrony.service.
root@kali:~# /etc/init.d/chrony status●
 chrony.service - chrony, an NTP client/server
     Loaded: loaded (/lib/systemd/system/chrony.service; disabled; vendor preset: disabled)
     Active: active (running) since Sat 2021-10-09 15:07:33 EDT; 2s ago
       Docs: man:chronyd(8)
             man:chronyc(1)
             man:chrony.conf(5)
    Process: 2938 ExecStart=/usr/sbin/chronyd $DAEMON_OPTS (code=exited, status=0/SUCCESS)
   Main PID: 2940 (chronyd)
      Tasks: 2 (limit: 2318)
```

```
     Memory: 1.8M
     CGroup: /system.slice/chrony.service
             |--2940 /usr/sbin/chronyd -F 1
             `-2941 /usr/sbin/chronyd -F 1
```

- Edite o arquivo de configuração, colocando agora o IP do servidor como responsável por gerar a hora, e então reinicie o serviço.

```
root@kali:~# vi /etc/chrony/chrony.conf
root@kali:~# /etc/init.d/chrony restart
```

- Verifique as informações sobre as fontes de hora que temos acesso.

```
root@kali:~# chronyc sources
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^? 10.11.10.150                  3   6     1     2  -4899us[-4899us] +/-   16ms
```

- Observe também os clientes associados ao servidor com o comando `chronyc clients`

```
[root@localhost joatham]# chronyc clients
Hostname                      NTP   Drop Int IntL Last     Cmd   Drop Int  Last
===============================================================================
10.11.10.43                     1      0   -   -    13       0      0   -     -
```

```
root@kali:~# chronyc sources -v

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current best, '+' = combined, '-' = not combined,
| /             'x' = may be in error, '~' = too variable, '?' = unusable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.           |  xxxx = adjusted offset,
||      Log2(Polling interval) --.      |          |  yyyy = measured offset,
||                                \     |          |  zzzz = estimated error.
||                                 |    |           \
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* 10.11.10.150                  3   6   377    33  +1266us[+2548us] +/-   15ms
```

Depois de fazer alterações no arquivo de configuração, reinicie o serviço `chronyd` e use `chronyc makestep` para ajustar manualmente o relógio do sistema:

```
[root@localhost joatham]# chronyc makestep
200 OK
```

Em seguida, use `chronyc tracking` para verificar se as alterações ocorreram.

```
shell [root@localhost joatham]# chronyc tracking Reference ID : C8A00008 (a.ntp.br)Stratum : 3
 Ref time (UTC): Sat Oct 09 19:41:37 2021 System time : 0.000000000 seconds slow of NTP time Last
  offset : +0.000164147 seconds RMS offset : 0.007542494 seconds Frequency : 487.924 ppm fast Residual
  freq : +0.005 ppm Skew : 0.441 ppm Root delay : 0.025843104 seconds Root dispersion : 0.002227175
  seconds Update interval : 64.3 seconds Leap status : Normal# Instale e atualize pacotes de
```

software da `Red Hat Network`, um repositório remoto ou do sistema de arquivos local ## Pontos de estudo para o exame Os candidatos ao exame `RHCSA` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Implantar, configurar e manter sistemas**
  - Agende tarefas usando at e cron
  - Inicie e pare serviços, além de configurar serviços para iniciar automaticamente durante a inicialização
  - Configure os sistemas para inicializar em um destino específico automaticamente
  - Configure clientes de serviço de tempo
  - **Instale e atualize pacotes de software da Red Hat Network, um repositório remoto ou do sistema de arquivos local**
  - Trabalhe com fluxos de módulo de pacote
  - Modifique o bootloader do sistema

Introducao

Um dos aspetos mais importantes num sistema é a capacidade de instalar e gerir diferentes tipos de pacotes, sendo esSas instalações as responsáveis por personalizar o servidor, dando-lhe a capacidade de agir perante às diferentes necessidades existentes. Num sistema RHEL, todos os pacotes de instalação são do tipo RPM. São usados utilitários como o yum e o rpm de modo a gerir esses mesmos pacotes.

Para começar esse tema, temos que perceber como é instalamos um novo programa/funcionalidade no sistema. Pense nisso!

De modo geral e simples, quando mandamos o sistema instalar um novo programa, o que ele realmente faz é comunicar com um repositório. Esse, por sua vez, está a gerir não só o software pretendido, mas todas as suas dependências, de modo a criar o mesmo de forma utilizável pelo sistema. Sendo assim, vamos observar e compreender estas diferentes etapas,

começando por analisar o que realmente é um repositório e como é que configuramos um no nosso sistema.

O papel dos repositórios

O *Yellowdog Updater, Modified*, é o utilitário padrão para gerenciar pacotes de software no *Red Hat Enterprise Linux*. No Fedora (a versão upstream do RHEL), o `Yum` foi substituído por `dnf`, mas a Red Hat decidiu manter o nome `Yum` para os lançamentos RHEL. Embora você esteja usando `yum`, por baixo do capô você está, de fato, usando `dnf`. E é por isso que às vezes você verá referências a `dnf` ou recursos `dnf`.

`Yum` é projetado para funcionar com repositórios que são depósitos de pacotes de software disponíveis. Repositórios tornam mais fácil manter sua máquina atualizada, uma vez que o mantenedor dos repositórios publica pacotes atualizados. Assim, sempre que você usa `yum` para instalar software, a versão mais recente é usada automaticamente.

Como um benefício adicional, o `yum` gerencia dependências de pacotes para que você não tenha que lidar com o inferno de resolvê-las. Quando um único pacote é instalado, ele contém informações sobre as dependências necessárias que o `yum` instalará automaticamente para você.

No *Red Hat Enterprise Linux*, você precisa registrar o sistema no Portal do Cliente Red Hat para obter acesso aos repositórios Red Hat. Se você não registrar o sistema, ficará sem repositório, mas poderá criar seu próprio repositório a partir da mídia de instalação.

Os repositórios são configurados no diretório `/etc/yum.repos.d/` como arquivos `.repo`. Cada arquivo pode conter vários repositórios, mas cada repositório deve ter pelo menos o seguinte conteúdo:

- **[label]** - Identifica o repositório específico.
- **name=** - Especifica o nome do repositório que você deseja usar.
- **baseurl=** - Contém a URL que aponta para os arquivos `.repo`. Pode ser `HTTP`, `FTP` ou um caminho de arquivo.

Visualize um repositório `RHEL`.

```
[root@localhost yum.repos.d]# vi redhat.repo

[rhel-8-for-x86_64-baseos-debug-rpms]
name = Red Hat Enterprise Linux 8 for x86_64 - BaseOS (Debug RPMs)
baseurl = https://cdn.redhat.com/content/dist/rhel8/$releasever/x86_64/baseos/debug
enabled = 0
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
```

```
sslcacert = /etc/rhsm/ca/redhat-uep.pem
sslclientkey = /etc/pki/entitlement/6816552860945528614-key.pem
sslclientcert = /etc/pki/entitlement/6816552860945528614.pem
metadata_expire = 86400
```

Como podemos observar, os pacotes nos repositórios da Internet são frequentemente assinados com uma chave GPG que permite verificar se foram alterados desde que o proprietário do repositório os publicou. Se por algum motivo a segurança do repositório for comprometida, a assinatura da chave GPG não corresponderá, e o comando yum chamará sua atenção sobre isso. Os pacotes assinados por GPG não são um requisito para repositórios internos ou locais.

Criando Seu Próprio Repositório

É bastante simples configurar seu próprio repositório, caso você não possa ou não queira registrar seu sistema RHEL. Você pode colocar seus próprios pacotes RPM (ou aqueles da mídia de instalação) em um diretório e publicar esse diretório como um repositório.

Se não estiver usando a mídia de instalação como fonte para seu próprio repositório, você precisará executar o comando `createrepo` dentro do diretório do repositório para gerar os metadados para os arquivos RPM.

Você pode criar repo a partir da mídia de instalação montando o arquivo ISO de forma persistente e criando o arquivo `.repo` necessário.

Crie o diretório vazio `/meu_repo` e adicione a seguinte linha na parte inferior do arquivo `/etc/fstab` para montar o arquivo ISO na próxima inicialização:

```
/path/to/file.iso    /repo    iso9660    defaults    0 0
```

Em seguida, monte o arquivo ISO:

```
[root@localhost ~]# mount /repo
mount: /repo: WARNING: device write-protected, mounted read-only.
[root@localhost ~]#
```

Crie o arquivo `/etc/yum.repos.d/mycustom.repo` e adicione o seguinte conteúdo:

```
[AppStream]
name=AppStream
```

```
baseurl=file:///repo/AppStream
gpgcheck=0

[BaseOS]
name=BaseOS
baseurl=file:///repo/BaseOS
gpgcheck=0
```

Verifique o repositório instalado usando `yum repolist`.

Trabalhando com Yum

Para usar repositórios, você precisa do comando `yum`. Abaixo, você encontrará uma visão geral das tarefas mais comuns do `yum`.

Command	Description
search	Pesquisa a string fornecida em nomes e resumos de pacotes.
[what]provides */name	Procura por arquivos específicos dentro de um pacote.
info	Retorna mais informações sobre um pacote.
install	Instala o pacote.
remove	Remove o pacote.
list [all]	Lista todos os pacotes instalados
group list	Lista os grupos de pacotes
group install [–with-optional]	Instala todos os pacotes de um grupo.
update	Atualiza pacotes ou pacotes especificos.
clean all	Remove todos os metadados armazenados.
history [undo]	Lista o histórico de comandos/desfaz um comando específico.

Para instalar um pacote, você precisa do nome exato; se não souber o nome exato, `yum search` pode ajudar a encontrá-lo. Lembre-se de que ele procura a string que você fornece nos nomes e resumos dos pacotes, então você não terá uma correspondência exata:

```
[root@localhost ~]# yum search 4linux
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:33:26 atrás em sáb 09 out 2021
     15:39:44 -03.
=================== Resumo Correspondeu: 4linux ====================
libv4l.x86_64 : Collection of video4linux support libraries
libv4l.i686 : Collection of video4linux support libraries
```

Temos tambem o comando `yum provides`, que pode ajudá-lo a encontrar arquivos dentro de um pacote, o que pode ser útil se você souber o nome do binário, por exemplo:

```
[root@localhost ~]# yum provides nmap
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:35:59 atrás em sáb 09 out 2021
     15:39:44 -03.
nmap-2:7.70-4.el8.x86_64 : Network exploration tool and security
                         : scanner
Repo        : rhel-8-for-x86_64-appstream-rpms
Resultado a partir de:
Fornecer    : nmap = 2:7.70-4.el8

nmap-2:7.70-5.el8.x86_64 : Network exploration tool and security
                         : scanner
Repo        : rhel-8-for-x86_64-appstream-rpms
Resultado a partir de:
Fornecer    : nmap = 2:7.70-5.el8
```

Podemos obter mais informações sobre um pacote use o comando `yum info`:

```
[root@localhost ~]# yum info nmap
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:37:15 atrás em sáb 09 out 2021
     15:39:44 -03.
Pacotes disponíveis
Nome         : nmap
Epoch        : 2
Versão       : 7.70
Lançamento   : 5.el8
Arquitetura  : x86_64
Tamanho      : 5.8 M
Origem       : nmap-7.70-5.el8.src.rpm
Repositório  : rhel-8-for-x86_64-appstream-rpms
Resumo       : Network exploration tool and security scanner
URL          : http://nmap.org/
Licença      : Nmap
Descrição    : Nmap is a utility for network exploration or security
             : auditing.  It supports ping scanning (determine which
             : hosts are up), many port scanning techniques
             : (determine what services the hosts are offering), and
             : TCP/IP fingerprinting (remote host operating system
             : identification). Nmap also offers flexible target and
             : port specification, decoy scanning, determination of
             : TCP sequence predictability characteristics,
             : reverse-identd scanning, and more. In addition to the
             : classic command-line nmap executable, the Nmap suite
             : includes a flexible data transfer, redirection, and
             : debugging tool (netcat utility ncat), a utility for
             : comparing scan results (ndiff), and a packet
             : generation and response analysis tool (nping).
```

O comando `yum list | less` nos mostrará uma lista de pacotes disponíveis e instalados. Se o

nome do repositório for mostrado, ou seja `@AppStream`, o pacote estará disponível para instalação naquele repositório. Se `@anaconda` for mostrado, então esse pacote já foi instalado:

```
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:39:17 atrás
em sáb 09 out 2021 15:39:44 -03.
Pacotes instalados
GConf2.x86_64                                   3.2.6-22.el8
                                    @AppStream

ModemManager.x86_64                             1.10.8-2.el8
                                    @anaconda

ModemManager-glib.x86_64                        1.10.8-2.el8
                                    @anaconda

NetworkManager.x86_64                           1:1.30.0-10.el
8_4                                 @rhel-8-for-x86_64-baseos-rpms

NetworkManager-adsl.x86_64                      1:1.30.0-10.el8_4
                                     @rhel-8-for-x86_64-baseos-rpms
NetworkManager-bluetooth.x86_64                 1:1.30.0-10.el
```

Os pacotes podem ser atualizados usando o comando `yum update`. A versão antiga de um pacote é substituída por uma nova versão, exceto para o pacote do kernel. O kernel mais novo é instalado junto com o kernel antigo para que você possa selecionar o kernel que deseja usar no `Grub` ao inicializar.

Para facilitar o gerenciamento de funcionalidades específicas em vez de pacotes específicos, podemos trabalhar com grupos de pacotes. Use o comando `yum groups list` para mostrar grupos de pacotes disponíveis e `yum groups info <groupname>` para ver quais pacotes estão no grupo especificado:

```
[root@localhost ~]# yum groups list
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:41:42 atrás em sáb 09 out 2021
     15:39:44 -03.
Grupos de Ambientes Disponíveis:
   Server
   Instalações Mínimas
   Workstation
   Máquina de Virtualização
   Sistema Operacional Personalizado
Grupos de Ambientes Instalados:
   Servidor com GUI
Grupos instalados:
   Gestão de Contentores
   Gestão sem Cabeça
   Ferramentas de Desenvolvimento
Grupos disponíveis:
   Ferramentas de Desenvolvimento RPM
   Desenvolvimento do núcleo .net
```

```
   Ferramentas do Sistema
   Ferramentas de Segurança
   Suporte Científico
   Ferramentas Administrativas gráficas
   Servidores de Rede
   Suporte a Smart Card
   Compatibilidade da Legacia UNIX.
[root@localhost ~]#
```

```
[root@localhost ~]# yum groups info "Ferramentas do Sistema"
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:43:05 atrás em sáb 09 out 2021
     15:39:44 -03.

Grupo: Ferramentas do Sistema
 Descrição: Este grupo é uma colecção de ferramentas para o sistema, tal como o cliente
      para se ligar a partilhas de SMB e as ferramentas para monitorizar o tráfego de rede.
 Pacotes padrão:
   NetworkManager-libreswan
   chrony
   cifs-utils
   libreswan
   nmap
   openldap-clients
   samba-client
   setserial
   tigervnc
   tmux
   xdelta
   zsh
 Pacotes opcionais:
   PackageKit-command-not-found
   aide
   amanda-client
   arpwatch
   autofs
   chrpath
   convmv
   createrepo_c
   environment-modules
   freerdp
   fuse
   gpm
   gssdp
   gupnp
   hardlink
   iotop
   lzop
   mc
   mrtg
   mtx
   net-snmp-utils
   oddjob
   oddjob-mkhomedir
   pmdk-convert
   rear
   speech-dispatcher
   speech-dispatcher-espeak-ng
   sysstat
```

```
   wireshark
   x3270-x11
[root@localhost ~]#
```

Você pode usar o comando `yum group install "System Tools"` para instalar os pacotes padrão dentro desse grupo. Se você também precisar dos Pacotes Opcionais, use o o comando `yum group install --with-optional "System Tools"`. Os grupos ocultos podem ser revelados usando o comando `yum groups list hidden`, estes são subgrupos de grupos específicos.

Consultando Pacotes de Software com RPM

Existem dois motivos pelos quais você não deve usar o comando `rpm` para gerenciar pacotes de software.

1. O `Yum` se encarrega de resolver as dependências do pacote para você, enquanto `rpm` não o faz.
2. Existem dois bancos de dados de pacote em um sistema `RHEL`, o banco de dados `YUM` e o banco de dados `RPM`. Quando você instala pacotes via `yum`, o banco de dados `YUM` é atualizado primeiro e as informações são sincronizadas com o banco de dados `RPM`. A instalação de pacotes com `RPM` atualizará apenas o banco de dados `RPM`. Isso não significa que o `RPM` não seja útil. Se você baixou um pacote `RPM`, ainda pode instalá-lo via comando `yum install package .rpm`. Mais importante ainda, o comando `rpm` nos permite obter mais informações sobre os pacotes:

Podemos usar o comando `rpm -qa` para mostrar uma lista de todos os softwares instalados no sistema, semelhante ao comando `yum list installed`. Podemos usar `grep` neste comando para descobrir nomes de pacotes específicos: `rpm -qa | grep utils`

```
[root@localhost ~]# rpm -qa | grep utils
poppler-utils-20.11.0-2.el8_4.1.x86_64
xorg-x11-utils-7.5-28.el8.x86_64
libpath_utils-0.2.1-39.el8.x86_64
libdb-utils-5.3.28-42.el8_4.x86_64
iprutils-2.4.19-1.el8.x86_64
xorg-x11-font-utils-7.5-40.el8.x86_64
patchutils-0.3.4-10.el8.x86_64
```

Vamos descobrir mais sobre o uso do pacote php-common com o comando `rpm -qi`:

```
[root@localhost ~]# rpm -qi binutils
Name        : binutils
Version     : 2.30
```

```
Release     : 93.el8
Architecture: x86_64
Install Date: sex 01 out 2021 16:30:56 -03
Group       : Unspecified
Size        : 24926705
License     : GPLv3+
Signature   : RSA/SHA256, qua 03 mar 2021 08:52:25 -03, Key ID 199e2f91fd431d51
Source RPM  : binutils-2.30-93.el8.src.rpm
Build Date  : seg 22 fev 2021 13:35:38 -03
Build Host  : x86-vm-15.build.eng.bos.redhat.com
Relocations : (not relocatable)
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Vendor      : Red Hat, Inc.
URL         : https://sourceware.org/binutils
Summary     : A GNU collection of binary utilities
Description :
Binutils is a collection of binary utilities, including ar (for
creating, modifying and extracting from archives), as (a family of GNU
assemblers), gprof (for displaying call graph profile data), ld (the
GNU linker), nm (for listing symbols from object files), objcopy (for
copying and translating object files), objdump (for displaying
information from object files), ranlib (for generating an index for
the contents of an archive), readelf (for displaying detailed
information about binary files), size (for listing the section sizes
of an object or archive file), strings (for listing printable strings
from files), strip (for discarding symbols), and addr2line (for
converting addresses to file and line).
```

Podemos listar os arquivos dentro do pacote usando o comando `rpm -ql`:

```
[root@localhost ~]# rpm -ql binutils
/usr/bin/addr2line
/usr/bin/ar
/usr/bin/as
/usr/bin/c++filt
/usr/bin/dwp
/usr/bin/elfedit
/usr/bin/gprof
/usr/bin/ld
/usr/bin/ld.bfd
/usr/bin/ld.gold
/usr/bin/nm
```

Ou podemos listar apenas a documentação usando o comando `rpm -qd`, ou os arquivos de configuração usando o comando `rpm -qc`:

```
[root@localhost ~]# rpm -qd binutils
/usr/share/doc/binutils/README
/usr/share/info/as.info.gz
/usr/share/info/bfd.info.gz
/usr/share/info/binutils.info.gz
/usr/share/info/gprof.info.gz
/usr/share/info/ld.info.gz
```

```
/usr/share/info/standards.info.gz
/usr/share/man/man1/addr2line.1.gz
/usr/share/man/man1/ar.1.gz
/usr/share/man/man1/as.1.gz
/usr/share/man/man1/c++filt.1.gz
/usr/share/man/man1/elfedit.1.gz
/usr/share/man/man1/gprof.1.gz
/usr/share/man/man1/ld.1.gz
/usr/share/man/man1/nm.1.gz
/usr/share/man/man1/objcopy.1.gz
/usr/share/man/man1/objdump.1.gz
/usr/share/man/man1/ranlib.1.gz
/usr/share/man/man1/readelf.1.gz
/usr/share/man/man1/size.1.gz
/usr/share/man/man1/strings.1.gz
/usr/share/man/man1/strip.1.gz
```

Se você tiver um nome de arquivo e quiser saber a qual pacote ele pertence, use o comando `rpm -qf`:

```
[root@localhost ~]# rpm -qf /bin/bash
bash-4.4.20-1.el8_4.x86_64
[root@localhost ~]# rpm -qf /bin/lsblk
util-linux-2.32.1-27.el8.x86_64
```

Todas as consultas acima foram usadas no banco de dados `RPM` e o que estávamos consultando eram pacotes instalados. Às vezes, faz sentido consultar um arquivo de pacote `RPM` antes de instalá-lo; nesse caso, precisamos adicionar a opção `-p` - além de qualquer uma das opções mencionadas anteriormente. Podemos usar o comando `yumdownloader` para baixar um pacote específico de nosso repositório para executar uma consulta `RPM` nele antes de instalá-lo:

```
[root@localhost ~]# yum whatprovides */yumdownloader
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 0:02:22 atrás em sáb 09 out 2021
     18:51:35 -03.
dnf-utils-4.0.2.2-3.el8.noarch : Yum-utils CLI compatibility layer
Repo        : rhel-8-for-x86_64-baseos-rpms
Resultado a partir de:
Nome de arquivo    : /usr/bin/yumdownloader

yum-utils-4.0.8-3.el8.noarch : Yum-utils CLI compatibility layer
Repo        : rhel-8-for-x86_64-baseos-rpms
Resultado a partir de:
Nome de arquivo    : /usr/bin/yumdownloader

yum-utils-4.0.12-3.el8.noarch : Yum-utils CLI compatibility layer
Repo        : rhel-8-for-x86_64-baseos-rpms
Resultado a partir de:
Nome de arquivo    : /usr/bin/yumdownloader

yum-utils-4.0.12-4.el8_2.noarch : Yum-utils CLI compatibility layer
```

```
Repo        : rhel-8-for-x86_64-baseos-rpms
Resultado a partir de:
Nome de arquivo    : /usr/bin/yumdownloader
```

Um breve resumo do que podera cair na prova quanto as flags do RPM

Comando	Descrição
rpm -qf	Use um nome de arquivo para encontrar o pacote RPM específico ao qual o arquivo pertence
rpm -ql	Fornece uma lista de arquivos dentro do pacote RPM
rpm -qi	Forneça informações sobre o pacote
rpm -qd	Mostra toda a documentação disponível no pacote
rpm -qc	Mostrar todos os arquivos de configuração
rpm -q -scripts	Mostra os scripts que são usados no pacote
rpm -qp	Consultar arquivos .rpm individuais em vez do banco de dados RPM
rpm -qR	Mostrar dependências de pacote
rpm -V	Mostra quais partes de um pacote foram alteradas desde a instalação.
rpm -Va	Verifica todos os pacotes instalados e mostra qual parte do pacote foi alterada desde a instalação.
rpm -qa	Lista todos os pacotes instalados

21

Trabalhar com fluxos de módulo de pacotes

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Implantar, configurar e manter sistemas**
  - Agende tarefas usando at e cron
  - Inicie e pare serviços, além de configurar serviços para iniciar automaticamente durante a inicialização
  - Configure os sistemas para inicializar em um destino específico automaticamente
  - Configure clientes de serviço de tempo
  - Instale e atualize pacotes de software da Red Hat Network, um repositório remoto ou do sistema de arquivos local
  - **Trabalhe com fluxos de módulo de pacote**
  - Modifique o bootloader do sistema

Introdução

A versao 8.0 do Red Hat traz à tona o conceito de *Application Streams*. Isso significa que diferentes versões dos aplicativos utilizados pelos usuários enviados com a distribuição agora são entregues ao mesmo tempo.

Eles podem ser atualizados com mais freqüência do que os pacotes principais do sistema operacional. Isso fornece aos usuarios uma maior flexibilidade para personalizar sua distribuicao Red Hat sem afetar a estabilidade subjacente da plataforma ou implementações específicas.

Tradicionalmente, manter e gerenciar diferentes versões de um pacote e suas depedências/-pacotes relacionados, significava manter repositórios diferentes para cada uma das versões diferente desses pacotes.

Para desenvolvedores que queriam a versão mais recente de um aplicativo e administradores que queriam a versão mais estável do aplicativo, isso criava uma situação complicada, para não dizer chata de gerenciar.

Bom, este processo é simplificado no Red Hat Enterprise Linux 8, que usa uma nova tecnologia chamada *Modularity*.

A modularidade permite que um único repositório hospede várias versões do pacote/aplicativo e suas dependências.

A titulo de conhecimento, é importante saber que tanto no CentOS quanto no RHEL 8, a distribuição dos pacotes são implementadas por meio de dois repositórios de software principais: **BaseOS** e **Application Stream (AppStream)**.

BaseOS

O conteúdo do repositório BaseOS se destina a fornecer o conjunto básico da funcionalidade do sistema operacional subjacente, que fornece a base para todas as instalações. Este conteúdo está disponível no formato RPM e está sujeito a termos de suporte semelhantes aos das versões anteriores do Red Hat Enterprise Linux.

AppStream

O conteúdo do repositório AppStream inclui aplicativos adicionais do espaço do usuário, linguagens de tempo de execução e bancos de dados para suportar as cargas de trabalho e casos de uso variados. Além de pacotes RPM individuais, o repositório AppStream contém módulos.

Modulos

Um módulo é um conjunto de pacotes RPM que representa um componente e geralmente são instalados juntos. Um módulo típico contém pacotes com um aplicativo, pacotes com bibliotecas de dependência específicas do aplicativo, pacotes com documentação para o aplicativo e pacotes com utilitários auxiliares. São organizados em suas sessões:

- **Streams** - organização do conteúdo por versão.
- **Perfis** - organização do conteúdo por propósito.

Fluxos de módulo

Os fluxos de módulo são filtros que podem ser imaginados como repositórios virtuais no repositório físico AppStream. Tais fluxos representam versões dos componentes AppStream. Cada um dos streams recebe atualizações de forma independente.

Os fluxos de módulo podem estar ativos ou inativos. Os fluxos ativos fornecem ao sistema acesso aos pacotes RPM dentro do fluxo de módulo específico, permitindo a instalação da respectiva versão do componente. Os fluxos estão ativos se marcados como padrão ou se forem explicitamente ativados por uma ação do usuário.

Apenas um fluxo de um módulo específico pode estar ativo em um determinado momento. Portanto, apenas uma versão de um componente pode ser instalada em um sistema.

Cada módulo pode ter um fluxo padrão. Os fluxos padrão facilitam o consumo de pacotes RHEL da maneira usual, sem a necessidade de aprender sobre os módulos. O fluxo padrão está ativo, a menos que todo o módulo tenha sido desabilitado ou outro fluxo desse módulo seja habilitado.

Certos fluxos de módulo dependem de outros fluxos de módulo. Por exemplo, os fluxos de módulo perl-App-cpanminus , perl-DBD-MySQL , perl-DBD-Pg , perl-DBD-SQLite , perl-DBI , perl-YAML e freeradius dependem de certos fluxos de módulo perl .

Para selecionar um fluxo específico para um aplicativo de usuário runtime ou um aplicativo de desenvolvedor, considere o seguinte:

- Funcionalidade necessária e quais versões de componentes a suportam
- Compatibilidade
- Duração do ciclo de vida e seu plano de atualização.

Perfis de módulo

Um perfil é uma lista de pacotes recomendados a serem instalados juntos para um caso de uso específico, como para um servidor, cliente, desenvolvimento, instalação mínima ou outro. Essas listas de pacotes podem conter pacotes fora do fluxo do módulo, geralmente do repositório BaseOS ou das dependências do fluxo.

Instalar pacotes usando um perfil é uma ação única fornecida para a conveniência do usuário. Isso não impede a instalação ou desinstalação de qualquer um dos pacotes fornecidos pelo

módulo. Também é possível instalar pacotes usando vários perfis do mesmo fluxo de módulo sem quaisquer etapas preparatórias adicionais.

Cada fluxo de módulo pode ter qualquer número de perfis, incluindo nenhum. Para qualquer fluxo de módulo fornecido, alguns de seus perfis podem ser marcados como padrão e, em seguida, usados para ações de instalação de perfil quando nenhum perfil é especificado explicitamente. No entanto, a existência de um perfil padrão para um fluxo de módulo não é necessária.

Exemplo de perfis de módulo `httpd`

O módulo `httpd`, que fornece o servidor web Apache, oferece os seguintes perfis para instalação:

- common - uma implantação reforçada e pronta para produção, o perfil padrão
- devel - os pacotes necessários para fazer modificações no httpd
- minimal - o menor conjunto de pacotes que fornecerá um servidor da web em execução.

Hands on

- Checando o modelo tradicional dos pacotes

```
[root@localhost joatham]# yum group list
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 1:21:25 atrás em qui 07 out 2021
     22:42:38 -03.
Grupos de Ambientes Disponíveis:
   Server
   Instalações Mínimas
   Workstation
   Máquina de Virtualização
   Sistema Operacional Personalizado
Grupos de Ambientes Instalados:
   Servidor com GUI
Grupos instalados:
   Gestão de Contentores
   Gestão sem Cabeça
   Ferramentas de Desenvolvimento
Grupos disponíveis:
   Ferramentas de Desenvolvimento RPM
   Desenvolvimento do núcleo .net
   Ferramentas do Sistema
   Ferramentas de Segurança
   Suporte Científico
   Ferramentas Administrativas gráficas
   Servidores de Rede
   Suporte a Smart Card
   Compatibilidade da Legacia UNIX.
```

- Listando todos os modulos disponiveis para todos os aplicativos.

```
[root@localhost joatham]# yum module list
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 1:26:48 atrás em qui 07 out 2021
     22:42:38 -03.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name                Stream          Profiles Summary
389-ds              1.4                      389 Directory Server (base)
ant                 1.10 [d]        common [ Java build tool
                                    d]
freeradius          3.0 [d]         server [ High-performance and highly configurable free
      RADIUS serve
                                    d]       r
gimp                2.8 [d]         common [ gimp module
                                    d], deve
                                    l
go-toolset          rhel8 [d]       common [ Go
                                    d]
httpd               2.4 [d]         common [ Apache HTTP Server
                                    d], deve
                                    l, minim
                                    al
idm                 DL1             adtrust, The Red Hat Enterprise Linux Identity
     Management system mo
                                     client, dule
                                     common
                                    [d], dns
                                    , server
```

- Entendendo o conceito de módulos

```
shell [root@localhost joatham]# yum module list | grep -i postgres perl-DBD-Pg 3.7 [d] common [
 d] A PostgreSQL interface for Perl postgresql 9.6 client, server [d] PostgreSQL server and client
  module postgresql 10 [d] client, server [d] PostgreSQL server and client module postgresql 12
  client, server [d] PostgreSQL server and client module postgresql 13 client, server [d] PostgreSQL
  server and client module
```

```
[root@localhost joatham]# yum module list postgresql
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 1:34:51 atrás em qui 07 out 2021
     22:42:38 -03.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name              Stream         Profiles                   Summary
postgresql        9.6            client, server [d]         PostgreSQL server and client
     module
postgresql        10 [d]         client, server [d]         PostgreSQL server and client
     module
postgresql        12             client, server [d]         PostgreSQL server and client
     module
postgresql        13             client, server [d]         PostgreSQL server and client
     module

Sugestão: [d] padrão, [e] habilitado, [x] desabilitado, [i] instalado
```

- Verificando os profiles dos módulos dos aplicativos

```
[root@localhost joatham]# yum module info postgresql --profile | less

Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 1:40:08 atrás em qui 07 out 2021
     22:42:38 -03.
Name   : postgresql:10:8020020200825115746:4cda2c84:x86_64
client : postgresql
server : postgresql-server

Name   : postgresql:10:8030020201201142418:229f0a1c:x86_64
client : postgresql
server : postgresql-server

Name   : postgresql:10:8040020210602185500:522a0ee4:x86_64
client : postgresql
server : postgresql-server

Name   : postgresql:10:820190104140132:9edba152:x86_64
client : postgresql
server : postgresql-server

Name   : postgresql:12:8010120191120141335:e4e244f9:x86_64
client : postgresql
server : postgresql-server
```

- Se quiséssemos instalar o modulo 9.6 do postgresql, precisaríamos executar o seguinte comando:

```
[root@localhost joatham]# yum module enable postgresql:9.6
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 1:44:26 atrás em qui 07 out 2021
     22:42:38 -03.
Dependências resolvidas.
================================================================================================================

 Pacote                  Arquitetura            Versão                     Repositório
                 Tamanho
================================================================================================================

Ativando Fluxos de Módulos:
 postgresql                                     9.6

Resumo da transação
================================================================================================================


Correto? [s/N]: s
Concluído!
```

```
[root@localhost joatham]# yum module list postgresql
Updating Subscription Management repositories.
```

```
Última verificação de data de vencimento de metadados: 1:46:56 atrás em qui 07 out 2021
     22:42:38 -03.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name               Stream          Profiles                   Summary
postgresql         9.6 [e]         client, server [d]         PostgreSQL server and client
     module
postgresql         10 [d]          client, server [d]         PostgreSQL server and client
     module
postgresql         12              client, server [d]         PostgreSQL server and client
     module
postgresql         13              client, server [d]         PostgreSQL server and client
     module

Sugestão: [d] padrão, [e] habilitado, [x] desabilitado, [i] instalado
```

- Instalando o módulo habilitado

```
[root@localhost joatham]# yum install postgresql
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 1:51:14 atrás em qui 07 out 2021
     22:42:38 -03.
Dependências resolvidas.
================================================================================================================

 Pacote      Arq.    Versão                                    Repositório
                            Tamanho
================================================================================================================

Instalando:
 postgresql  x86_64  9.6.22-1.module+el8.4.0+11244+beebcf7e     rhel-8-for-x86_64-appstream
      -rpms  1.4 M
Instalando dependências:
 libpq       x86_64  13.3-1.el8_4                              rhel-8-for-x86_64-appstream
      -rpms  197 k

Resumo da transação
================================================================================================================

Instalar  2 Pacotes

Tamanho total do download: 1.6 M
Tamanho depois de instalado: 5.7 M
Correto? [s/N]: N
```

- Habilitando outro módulo para o pacote postgresql

```
[root@localhost joatham]# yum module enable postgresql:12
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 1:53:28 atrás em qui 07 out 2021
     22:42:38 -03.
Dependências resolvidas.
The operation would result in switching of module 'postgresql' stream '9.6' to stream '12'
Erro: It is not possible to switch enabled streams of a module.
```

```
It is recommended to remove all installed content from the module, and reset the module
     using 'yum module reset <module_name>' command. After you reset the module, you can
     install the other stream.
```

> ERRO!!! nao 'e possivel ter mais de um fluxo ao mesmo tempo.

- Resetando o módulo como solicitado

```
[root@localhost joatham]# yum module reset postgresql
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 1:58:21 atrás em qui 07 out 2021
     22:42:38 -03.
Dependências resolvidas.
================================================================================================================

 Pacote                  Arquitetura            Versão                    Repositório
                  Tamanho
================================================================================================================

Redefinindo módulos:
 postgresql

Resumo da transação
================================================================================================================


Correto? [s/N]: s
Concluído!
```

```
[root@localhost joatham]# yum module list postgresql
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:00:13 atrás em qui 07 out 2021
     22:42:38 -03.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name               Stream         Profiles                  Summary
postgresql         9.6            client, server [d]        PostgreSQL server and client
     module
postgresql         10 [d]         client, server [d]        PostgreSQL server and client
     module
postgresql         12             client, server [d]        PostgreSQL server and client
     module
postgresql         13             client, server [d]        PostgreSQL server and client
     module

Sugestão: [d] padrão, [e] habilitado, [x] desabilitado, [i] instalado
```

```
[root@localhost joatham]# yum module enable postgresql:12
Updating Subscription Management repositories.
```

```
Última verificação de data de vencimento de metadados: 2:03:35 atrás em qui 07 out 2021
     22:42:38 -03.
Dependências resolvidas.
================================================================================================================

 Pacote                  Arquitetura            Versão                     Repositório
                  Tamanho
================================================================================================================

Ativando Fluxos de Módulos:
 postgresql                                     12

Resumo da transação
================================================================================================================


Correto? [s/N]: s
Concluido
```

- Trabalhando com perfis diferentes

```
[root@localhost joatham]# yum module list httpd
Updating Subscription Management repositories.š
Ãltima verificaÃ§Ã£o de data de vencimento de metadados: 2:13:05 atrÃ¡s em qui 07 out 2021
     22:42:38 -03.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name              Stream             Profiles                              Summary
httpd             2.4 [d]            common [d], devel, minimal            Apache HTTP
     Server

SugestÃ£o: [d] padrÃ£o, [e] habilitado, [x] desabilitado, [i] instalado
```

```
[root@localhost joatham]# yum module install httpd/minimal
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:13:57 atrás em qui 07 out 2021
     22:42:38 -03.
Dependências resolvidas.
================================================================================================================

 Pacote            Arq.   Versão                                    Repositório
                          Tamanho
================================================================================================================

Instalando grupo/pacotes do módulo:
 httpd             x86_64 2.4.37-39.module+el8.4.0+9658+b87b2deb rhel-8-for-x86_64-
      appstream-rpms 1.4 M
Instalando dependências:
 apr               x86_64 1.6.3-11.el8                           rhel-8-for-x86_64-
      appstream-rpms 125 k
 apr-util          x86_64 1.6.1-6.el8                            rhel-8-for-x86_64-
      appstream-rpms 105 k
 httpd-filesystem  noarch 2.4.37-39.module+el8.4.0+9658+b87b2deb rhel-8-for-x86_64-
      appstream-rpms  38 k
 httpd-tools       x86_64 2.4.37-39.module+el8.4.0+9658+b87b2deb rhel-8-for-x86_64-
      appstream-rpms 106 k
```

```
 mod_http2         x86_64 1.15.7-3.module+el8.4.0+8625+d397f3da  rhel-8-for-x86_64-
      appstream-rpms 154 k
 redhat-logos-httpd
                   noarch 84.4-1.el8                             rhel-8-for-x86_64-baseos-
                        rpms     29 k
Instalando dependências fracas:
 apr-util-bdb      x86_64 1.6.1-6.el8                            rhel-8-for-x86_64-
      appstream-rpms  25 k
 apr-util-openssl  x86_64 1.6.1-6.el8                            rhel-8-for-x86_64-
      appstream-rpms  27 k
Instalando perfis de módulo:
 httpd/minimal
Ativando Fluxos de Módulos:
 httpd                    2.4

Resumo da transação
================================================================================================================

Instalar  9 Pacotes

Tamanho total do download: 2.0 M
Tamanho depois de instalado: 5.4 M
Correto? [s/N]:
```

```
[root@localhost joatham]# yum module install httpd/devel
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 2:16:04 atrás em qui 07 out 2021
     22:42:38 -03.
Dependências resolvidas.
================================================================================================================

 Pacote            Arq.   Versão                                 Repositório
                           Tamanho
================================================================================================================

Instalando grupo/pacotes do módulo:
 httpd             x86_64 2.4.37-39.module+el8.4.0+9658+b87b2deb rhel-8-for-x86_64-
      appstream-rpms 1.4 M
 httpd-devel       x86_64 2.4.37-39.module+el8.4.0+9658+b87b2deb rhel-8-for-x86_64-
      appstream-rpms 221 k
 httpd-filesystem  noarch 2.4.37-39.module+el8.4.0+9658+b87b2deb rhel-8-for-x86_64-
      appstream-rpms  38 k
 httpd-tools       x86_64 2.4.37-39.module+el8.4.0+9658+b87b2deb rhel-8-for-x86_64-
      appstream-rpms 106 k
Instalando dependências:
 apr               x86_64 1.6.3-11.el8                           rhel-8-for-x86_64-
      appstream-rpms 125 k
 apr-devel         x86_64 1.6.3-11.el8                           rhel-8-for-x86_64-
      appstream-rpms 246 k
 apr-util          x86_64 1.6.1-6.el8                            rhel-8-for-x86_64-
      appstream-rpms 105 k
 apr-util-devel    x86_64 1.6.1-6.el8                            rhel-8-for-x86_64-
      appstream-rpms  86 k
 cyrus-sasl-devel  x86_64 2.1.27-5.el8                           rhel-8-for-x86_64-baseos-
      rpms    128 k
 expat-devel       x86_64 2.2.5-4.el8                            rhel-8-for-x86_64-baseos-
      rpms     55 k
 libdb-devel       x86_64 5.3.28-42.el8_4                        rhel-8-for-x86_64-
      appstream-rpms  47 k
```

```
 mod_http2         x86_64 1.15.7-3.module+el8.4.0+8625+d397f3da  rhel-8-for-x86_64-
      appstream-rpms 154 k
 openldap-devel    x86_64 2.4.46-17.el8_4                        rhel-8-for-x86_64-baseos-
      rpms    811 k
 redhat-logos-httpd
                   noarch 84.4-1.el8                             rhel-8-for-x86_64-baseos-
                        rpms     29 k
Instalando dependências fracas:
 apr-util-bdb      x86_64 1.6.1-6.el8                            rhel-8-for-x86_64-
      appstream-rpms  25 k
 apr-util-openssl  x86_64 1.6.1-6.el8                            rhel-8-for-x86_64-
      appstream-rpms  27 k
Instalando perfis de módulo:
 httpd/devel
Ativando Fluxos de Módulos:
 httpd                    2.4

Resumo da transação
================================================================================================================

Instalar  16 Pacotes

Tamanho total do download: 3.6 M
Tamanho depois de instalado: 12 M
Correto? [s/N]:
```

22

Modificar o carregador de inicialização do sistema.

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Implantar, configurar e manter sistemas**
  - Agende tarefas usando at e cron
  - Inicie e pare serviços, além de configurar serviços para iniciar automaticamente durante a inicialização
  - Configure os sistemas para inicializar em um destino específico automaticamente
  - Configure clientes de serviço de tempo
  - Instale e atualize pacotes de software da Red Hat Network, um repositório remoto ou do sistema de arquivos local
  - Trabalhe com fluxos de módulo de pacote
  - **Modifique o bootloader do sistema**

Introducao

Para controlar sua máquina, o componente principal do sistema operacional, no caso o kernel, deve ser carregado por um programa chamado *bootloader* (carregador de inicialização). O

booatloader, por sua vez, é carregado por um firmware pré-instalado, como a BIOS ou a UEFI. O carregador de inicialização pode ser personalizado para passar parâmetros para o kernel, como a partição que contém o sistema de arquivos raiz ou em qual modo o sistema operacional deve ser executado. Uma vez carregado, o kernel dá seguimento ao processo de inicialização, identificando e configurando o hardware. Por fim, o kernel chama o utilitário responsável por iniciar e gerenciar os serviços do sistema.

O carregador de inicialização mais popular para Linux na arquitetura x86 é o GRUB (Grand Unified Bootloader). Assim que é chamado pela BIOS ou pela UEFI, o GRUB exibe uma lista de sistemas operacionais disponíveis para inicialização. Às vezes, a lista não aparece automaticamente, mas ela pode ser invocada pressionando Shift enquanto o GRUB está sendo chamado pela BIOS. Nos sistemas UEFI, a tecla a ser usada é Esc.

No menu do GRUB, é possível escolher qual dos kernels instalados deve ser carregado e passar novos parâmetros para ele. A maioria dos parâmetros do kernel segue o padrão `opção=valor`. Alguns dos parâmetros mais úteis do kernel são:

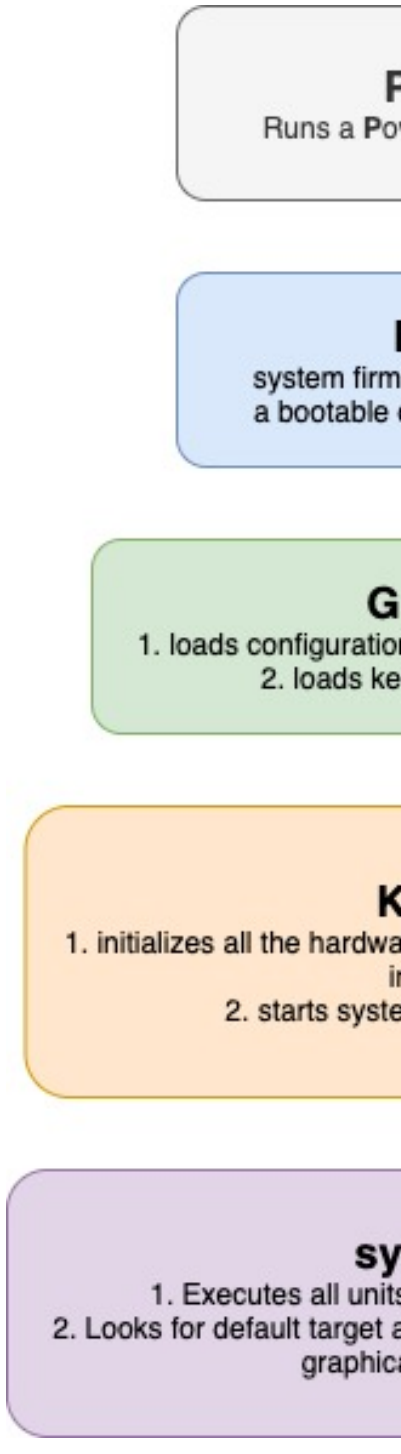
- **acpi** - Ativa/desativa o suporte a ACPI. acpi=off desabilita o suporte a ACPI.
- **init** - Define um iniciador de sistema alternativo. Por exemplo, init=/bin/bash define o shell Bash como iniciador. Assim, uma sessão do shell será iniciada logo após o processo de inicialização do kernel.
- **systemd.unit** - Define o destino do systemd a ser ativado. Por exemplo, systemd.unit=graphical.target. O systemd também aceita os níveis de execução numéricos definidos para SysV. Para ativar o nível de execução 1, por exemplo, é necessário apenas incluir o número 1 ou a letra S (abreviação de "single") como parâmetro do kernel.
- **mem** - Define a quantidade de RAM disponível para o sistema. Este parâmetro é útil para limitar a RAM disponível para cada convidado em uma máquina virtual. Assim, mem=512M limita a 512 megabytes a quantidade de RAM disponível para um sistema convidado em particular.
- **maxcpus** - Limita o número de processadores (ou núcleos de processador) visíveis ao sistema em máquinas multiprocessador simétricas. Também é útil para máquinas virtuais. Um valor de 0 desativa o suporte a máquinas multiprocessador e tem o mesmo efeito do parâmetro do kernel nosmp. O parâmetro maxcpus=2 limita a dois o número de processadores disponíveis para o sistema operacional.
- **quiet** - Oculta a maioria das mensagens de inicialização.
- **vga** - Seleciona um modo de vídeo. O parâmetro vga=ask mostra uma lista dos modos disponíveis a escolher.
- **root** - Define a partição raiz, diferente da que está configurada no bootloader. Por exemplo, root=/dev/sda3.
- **rootflags** - Opções de montagem para o arquivo de sistemas raiz.
- **ro** - Torna somente para leitura a montagem inicial do arquivo de sistemas raiz.

rw - Permite escrever no arquivo de sistemas raiz durante a montagem inicial.

Geralmente não é necessário alterar os parâmetros do kernel, mas isso pode ser útil para detectar e resolver problemas relacionados ao sistema operacional. Os parâmetros do kernel devem ser adicionados ao arquivo `/etc/default/grub` na linha `GRUB_CMDLINE_LINUX`, para que persistam após a inicialização.

É necessário gerar um novo arquivo de configuração para o carregador de inicialização a cada vez que `/etc/default/grub` é alterado, o que é feito com o comando `grub-mkconfig -o /boot/grub /grub.cfg`. Quando o sistema operacional estiver rodando, os parâmetros do kernel usados para carregar a sessão ficam disponíveis para leitura no arquivo `/proc/cmdline`.

Num sistema RHEL8, as pastas responsáveis pela configuração do `GRUB2` estão espalhadas por outras diferentes pastas, sendo que abaixo ficam definidas as suas localizações e os seus propósitos: - `/boot/grub2/grub.cfg` – Esta pasta contém a configuração final do `GRUB2`. O arquivo não deve ser editado manualmente. - `/etc/grub2.cfg` – Esta pasta é apenas um link simbólico para pastas anteriores (`/boot/grub2/grub.cfg`). - `/etc/default/grub` – Esta pasta contém a lista das diferentes variáveis do `GRUB2`, sendo que os valores destas variáveis podem ser alterados livremente. - `/etc/sysconfig/grub` - Esta pasta é apenas um link simbólico para a pasta anterior (`/etc/default/grub`). - `/etc/grub.d/` - Neste diretório existem todos os diferentes de ficheiros que são internamente usados pelo GRUB2.



Uma imagem que retrata muito bem como acontece esse processo está abaixo:

Configurando GRUB2 usando a ferramenta grubby

Acreditamos que os objetivos da RHCSA se referem a esta ferramenta. A ferramenta `grubby` pode ser usada para ler informações e fazer alterações persistentes no arquivo `grub.cfg`. Ele permite, por exemplo, alterar as entradas do menu `GRUB` para especificar quais argumentos passar para um kernel na inicialização do sistema e alterar o kernel padrão. No Red Hat Enterprise Linux 8, o padrão é trabalhar com o `grubby` para editar arquivo de configuração GRUB 2 `grub.cfg`.

Hands On

Listando o kernel padrão

Para descobrir o nome do arquivo do kernel padrão, digite o seguinte comando:

```
[root@localhost joatham]# grubby --default-kernel
/boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64
```

Para descobrir o número do índice do kernel padrão, digite o seguinte comando:

```
[root@localhost joatham]# grubby --default-index
0
```

Visualizando a entrada do menu GRUB para um kernel

Para listar todas as entradas do menu do kernel, digite um comando da seguinte maneira:

```
[root@localhost joatham]# grubby --info=ALL
index=0
kernel="/boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/
     swap rhgb quiet $tuned_params"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-4.18.0-305.19.1.el8_4.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (4.18.0-305.19.1.el8_4.x86_64) 8.4 (Ootpa)"
id="b002bcbcf6d54646b9265924d6266e20-4.18.0-305.19.1.el8_4.x86_64"
index=1
kernel="/boot/vmlinuz-4.18.0-305.el8.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/
     swap rhgb quiet $tuned_params"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-4.18.0-305.el8.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (4.18.0-305.el8.x86_64) 8.4 (Ootpa)"
id="b002bcbcf6d54646b9265924d6266e20-4.18.0-305.el8.x86_64"
```

```
index=2
kernel="/boot/vmlinuz-0-rescue-b002bcbcf6d54646b9265924d6266e20"
args="ro crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/
     swap rhgb quiet"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-0-rescue-b002bcbcf6d54646b9265924d6266e20.img"
title="Red Hat Enterprise Linux (0-rescue-b002bcbcf6d54646b9265924d6266e20) 8.4 (Ootpa)"
id="b002bcbcf6d54646b9265924d6266e20-0-rescue"
```

Para visualizar a entrada do menu GRUB para um kernel específico, digite o seguinte comando:

```
[root@localhost joatham]# grubby --info /boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64
index=0
kernel="/boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/
     swap rhgb quiet $tuned_params"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-4.18.0-305.19.1.el8_4.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (4.18.0-305.19.1.el8_4.x86_64) 8.4 (Ootpa)"
id="b002bcbcf6d54646b9265924d6266e20-4.18.0-305.19.1.el8_4.x86_64"
```

Alterando a entrada de inicialização padrão

Para fazer uma mudança persistente no kernel designado como kernel padrão, use o seguinte comando `grubby`:

```
[root@localhost joatham]# grubby --set-default /boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64
The default is /boot/loader/entries/b002bcbcf6d54646b9265924d6266e20-4.18.0-305.19.1.el8_4.
     x86_64.conf with index 0 and kernel /boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64
```

ou

```
[root@localhost joatham]# grubby --set-default-index=1
The default is /boot/loader/entries/b002bcbcf6d54646b9265924d6266e20-4.18.0-305.el8.x86_64.
     conf with index 1 and kernel /boot/vmlinuz-4.18.0-305.el8.x86_64
```

```
[root@localhost joatham]# reboot
```

```
[root@localhost joatham]# grubby --info=DEFAULT
```

Adicionando e removendo argumentos de uma entrada do menu GRUB

A opção `--update-kernel` pode ser usada para atualizar uma entrada de menu quando usada em combinação com `--args` para adicionar novos argumentos e `--remove-arguments` para remover argumentos existentes. Essas opções aceitam uma lista separada por espaço entre aspas. O comando para adicionar e remover simultaneamente argumentos da entrada do menu GRUB tem o seguinte formato:

```
[root@localhost joatham]# grubby --info=DEFAULT
index=1
kernel="/boot/vmlinuz-4.18.0-305.el8.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/
     swap rhgb quiet $tuned_params"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-4.18.0-305.el8.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (4.18.0-305.el8.x86_64) 8.4 (Ootpa)"
id="b002bcbcf6d54646b9265924d6266e20-4.18.0-305.el8.x86_64"
```

```
[root@localhost joatham]# grubby --remove-args=quiet --update-kernel=DEFAULT
```

```
[root@localhost joatham]# grubby --info=DEFAULT
index=1
kernel="/boot/vmlinuz-4.18.0-305.el8.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/
     swap rhgb $tuned_params"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-4.18.0-305.el8.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (4.18.0-305.el8.x86_64) 8.4 (Ootpa)"
id="b002bcbcf6d54646b9265924d6266e20-4.18.0-305.el8.x86_64"
```

```
[root@localhost joatham]# reboot
```

```
[root@localhost joatham]# grubby --args=quiet --update-kernel=DEFAULT
```

```
[root@localhost joatham]# grubby --info=DEFAULT
index=1
kernel="/boot/vmlinuz-4.18.0-305.el8.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/
     swap rhgb $tuned_params quiet"
root="/dev/mapper/rhel-root"
initrd="/boot/initramfs-4.18.0-305.el8.x86_64.img $tuned_initrd"
title="Red Hat Enterprise Linux (4.18.0-305.el8.x86_64) 8.4 (Ootpa)"
id="b002bcbcf6d54646b9265924d6266e20-4.18.0-305.el8.x86_64"
```

Outra forma de modificar o gerenciador de boot

```
[root@localhost joatham]# cd /etc/default/
[root@localhost default]# ls
grub  useradd
[root@localhost default]# vi grub

GRUB_TIMEOUT=30
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.
     lvm.lv=rhel/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```

```
[root@localhost grub2]# cp grub.cfg /tmp/grub.cfg.bkp
[root@localhost grub2]# grub2-mkconfig -o ./grub.cfg
Generating grub configuration file ...
done
```

```
[root@localhost grub2]# diff grub.cfg /tmp/grub.cfg.bkp
65c65
<   set timeout=30
---
>   set timeout=5
69c69
<   set timeout=30
---
>   set timeout=5
```

```
[root@localhost joatham]# reboot
```

23

Configurar endereços IPv4 e IPv6

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar rede básica**
  - **Configurar endereços IPv4 e IPv6**
  - Configurar resolução de nome de host
  - Configurar os serviços de rede para iniciar automaticamente na inicialização
  - Restringir o acesso à rede usando firewall-cmd/firewall

Introdução

Cada casa tem um laptop, smartphone, relógio digital, dispositivo IoT, componente de automação residencial e outros dispositivos conectados à rede doméstica ou à Internet. Os dispositivos se comunicam por meio de vários protocolos de rede, sendo o TCP e o IP os protocolos mais usados. Cada dispositivo conectado à rede deve ter um endereço IP que identifica o dispositivo na rede.

O que é IP (Internet Protocol)?

Sendo humanos, nós nos identificamos e nos comunicamos usando nossos nomes. Da mesma forma, no mundo da computação, os dispositivos usam endereços IP para se identificar e interagir uns com os outros. Uma linguagem comum usada por todos os dispositivos de computação para se comunicarem uns com os outros é conhecida como protocolo. Semelhante às linguagens humanas, o protocolo também possui um conjunto de regras que formata e processa os dados.

O protocolo da Internet (IP) é um conjunto de regras que especifica o endereçamento e o roteamento dos dados entre computadores. É usado principalmente com protocolos de transporte de rede, como TCP e UDP.

A Internet existe hoje devido a esse padrão de endereçamento exclusivo. A IANA gerencia os intervalos de endereços IP para redes/sites que se conectam à Internet. No entanto, se executarmos uma infraestrutura de rede local isolada, podemos atribuir números de acordo com nossa preferência. Agora, vamos entrar em detalhes sobre a arquitetura e sua análise comparativa.

IPv4

A primeira versão principal do Protocolo da Internet (IP) é a versão 4 (IPv4). Ela usa um esquema de endereçamento de rede de `32 bits` que se divide em quatro números de `8 bits`, conhecidos como octetos. Por exemplo, `google.com` tem um endereço `IP` de `141.251.36.46`. Esses endereços IP podem ser configurados manualmente ou obtidos automaticamente por meio de um servidor `DHCP`.

Para verificar o status ao vivo do dispositivo remoto, podemos fazer uma sondagem `ICMP` para esse `IP` usando o comando `ping`:

```
ping -c 1 google.com
```

Os endereços `IP` são divididos em duas partes, rede e endereços de host para a criação de sub-redes. Os números da sub-rede ajudam a decidir a rede e as partes do host do IP. Além disso, o espaço IP disponível é dividido em cinco classes diferentes, conforme tabulado a seguir.

Classe de endereço	Intervalos de IP	Máscara de sub-rede	Nž de redes	Número de hosts por rede
Classe A	1.0.0.0 a 126.0.0.0	255.0.0.0	126	16.777.214
Classe B	128.0.0.0 a 191.255.0.0	255.255.0.0	16.282	65.534
Classe C	192.0.0.0 a 223.255.255.0	255.255.255.0	2.097.150	254
Classe D	224.0.0.0 a 239.255.255.255	Multicast		
Classe E	240.0.0.0 a 255.255.255.255	Pesquisa/Reservado /Experimental		

O cálculo da sub-rede envolve algumas rubricas matemáticas atrás da tela. Para facilitar nosso cálculo, podemos usar ferramentas como `ipcalc` ou `subnetcalc` para sub-redes IPv4. Os snippets abaixo mostram como usar as ferramentas:

Para identificar o IP da interface de rede, você pode usar comandos como `ifconfig`, `hostname` ou `ip`.

Existem endereços IP de rede para fins especiais, como `0.0.0.0` ou `127.0.0.1`. A primeira é a rota padrão ou rota quad-zero, enquanto a última é chamada de endereço de `loopback`.

IPv6

A versão 6 do protocolo da Internet é a versão atualizada. Ela tem como objetivo substituir a versão mais antiga (IPv4), que transportava 75% do tráfego total da Internet em 2018 (Fonte: Google IPv6 Stats).

O endereço IPv6 é de `128 bits` (16 bytes), usando `32 dígitos` hexadecimais, enquanto esses dígitos são divididos em `oito` grupos de `quatro` dígitos cada para facilitar o gerenciamento.

Existem algumas etapas básicas envolvidas na convenção de nomenclatura `IPv6`.

- Regra 1: todas as letras não diferenciam maiúsculas de minúsculas. Por exemplo, `ab41` é igual a `AB41`
- Regra 2: campos sucessivos com `0` podem ser visualizados como `::`, mas apenas uma vez em um endereçamento.

- Regra 3: representar zeros à esquerda em um campo é opcional. Por exemplo, `001a` é igual a `1a`

Por exemplo, vamos pegar o endereço IPv6, `45ab:0000:a179:0000:0000:c1c0:abcd:0876`.

Aplicar regra 1 => `45ab:0000:a179:0000:0000:c1c0:abcd:0876`.

Aplicar regra 2 => `45ab:0:a179:0:0:c1c0:abcd:876`.

Aplicar regra 3 => `45ab:0:a179::c1c0:abcd:876`

Existem três tipos de endereços IPv6: Unicast, Multicast e Anycast. O endereço unicast é a única interface de rede e os pacotes entregues a essa interface específica. Além disso, os endereços unicast têm níveis de escopo local (link-local) e global.

O endereço multicast são as interfaces de grupo para as quais os pacotes são entregues. O endereço anycast é a interface do grupo e o pacote entregue à interface mais próxima.

Poucos endereços conhecidos são tabulados a seguir.

] :: 1/128	Endereço de loopback
ff00 :: / 8	Endereços multicast
fe80 :: / 10	Endereços locais de link
2001 :: / 16	Endereços unicast IPv6 regulares
2002 :: / 16	Endereços Unicast 6to4

```
subnetcalc 2001:4860:4860::8888/64
```

Embora o IPv6 tenha inúmeras vantagens, ele não pode suplantar o IPv4. Ambas as versões do protocolo devem coexistir por algum tempo para uma migração perfeita.

Consequentemente, os provedores de serviço estão oferecendo um sistema de suporte de pilha dupla que possui uma interface de rede que pode entender pacotes IPv4 e IPv6. Existem poucos mecanismos de transição inteligente, nomeadamente encapsulamento IPv6, endereços IPv6 mapeados em IPv4 etc. O primeiro encapsula o pacote IPv6 em IPv4, enquanto o último mapeia os endereços IPv6 para IPv4 nas implementações de pilha dupla.

IPv4 vs IPv6 - Análise Comparativa Rápida

Recursos	Protocolo de Internet - Versão 4 IPv4	Protocolo de Internet - Versão 6 IPv6
Implantação e alocação	1981	1999
Comprimento	32 bits	128 bits
Espaço de Endereçamento	4,29 x 10 ^ 9	3,4 x 10 ^ 38
Formato	Decimal pontuado/[10.235.64.56]	Hexadecimal/[2400 :: 4]
Número de octetos	4	16
Tamanho do cabeçalho	Varia de 20 a 60 bytes	40 bytes
Classes	Cinco classes: Classe A, Classe B, Classe C, Classe D, Classe E	Nenhum
Recursos de segurança/autenticação e criptografia	Não disponível	Disponível
Checksum	Disponível	Não disponível
IPSec	Externo e opcional	Recurso embutido
Contagem de saltos	Indicado pelo campo TTL	Indicado pelo campo Hoplimit
Fragmentação	Executado pelo remetente e roteadores de encaminhamento	Feito apenas pelo remetente
Campos de Opção	Fornecido no cabeçalho IPv4	Não há campos opcionais, mas os cabeçalhos de extensão IPv6 estão disponíveis
Multicast	IGMP gerencia os membros do grupo multicast	MLD substitui o IGMP
Transmitir mensagem	Disponível	Não disponível. Multicast é usado
Mapeamento de IP para MAC	Protocolo de Resolução de Endereço	Protocolo de Descoberta de Vizinhos

Administrando Endereços

Como administrador do Linux, você tem várias ferramentas para usar e configurar suas conexões de rede, como: `nmtui`, `NetworkManager` com interface gráfica de usuário GNOME e, claro, `nmcli` (ferramenta de linha de comando do gerenciador de rede).

Temos visto muitos administradores usando `nmtui` para simplificar. No entanto, o uso do `nmcli` economiza seu tempo, dá a você confiança, pode usá-lo em scripts e é a primeira ferramenta a ser usada para solucionar problemas de rede do servidor Linux e trazer de volta rapidamente sua funcionalidade.

A sintaxe do comando `nmcli`:

```
# nmcli [OPTIONS] OBJECT {COMMAND | help}
```

`nmcli` é usado para criar, exibir, editar, excluir, ativar e desativar conexões de rede, bem como controlar e exibir o status do dispositivo de rede. As conexões são armazenadas em arquivos de configuração. O serviço `NetworkManager` deve estar em execução para gerenciar esses arquivos.

Compare as configurações nm com as diretivas ifcfg- * (IPv4)

nmcli con mod	arquivo ifcfg- *	Efeito
ipv4.method manual	BOOTPROTO=none	Endereço IPv4 configurado estaticamente
ipv4.method auto	BOOTPROTO=dhcp	Irá procurar as configurações de um servidor DHCPv4
ipv4.address “192.168.0.10/24”	IPADDR=192.168.0.10 PREFIX=24	Definir endereço IPv4 estático, prefixo de rede
ipv4.gateway 192.168.0.1	GATEWAY=192.168.0.1	Definir Gateway IPv4

nmcli con mod	arquivo ifcfg- *	Efeito
ipv4.dns 8.8.8.8	DNS1=8.8.8.8	Modifique /etc/resolv.conf para usar este servidor de nomes
ipv4.dns-search example.com	DOMAIN=example.com	Modifique `/etc/resolv.conf` para usar este domínio na diretiva de pesquisa
ipv4.ignore-auto-dns true	PEERDNS=no	Ignorar as informações do servidor DNS do servidor DHCP
connection.autoconnect yes	ONBOOT=yes	Ativar automaticamente esta conexão na inicialização
connection.id eth0	NAME=eth0	O nome desta conexão
connection.interface-name eth0	DEVICE=eth0	A conexão está ligada à interface de rede com este nome
802-3-ethernet.mac-address 08:00:27:4b:7a:80	HWADDR=08:00:27:4b:7a:80	A conexão está ligada à interface de rede com este endereço MAC

nmcli con mod	arquivo ifcfg- *	Efeito
ipv4.never-default no	DEFROUTE=yes	Nunca use o gateway da interface fornecida como gateway padrão.

Compare as configurações nm com as diretivas ifcfg- * (IPv6)

nmcli con mod	arquivo ifcfg- *	Efeito
ipv6.method manual	IPV6_AUTOCONF=no	IPv6 está configurado estaticamente
ipv6.method auto	IPV6_AUTOCONF=yes	Irá definir as configurações de rede usando SLAAC de anúncios de roteador.
ipv6.method dhcp	IPV6_AUTOCONF=no DHCPV6C=yes	
ipv6 . addresses “2001:db8::a/64 2001:db8::1”	IPV6ADDR=2001:db8::a/64 IPV6_DEFAULTGW=2001:db8::1	Define endereço IPv6 estático e gateway
ipv6.dns . . .	DNS0=. . .	Modifique /etc/resolv.conf para usar este servidor de nomes
ipv6.dns-search example.com	DOMAIN=example.com	Modifique /etc/resolv.conf para usar este domínio na diretiva de pesquisa
ipv6.ignore-auto-dns true	IPV6_PEERDNS=no	Ignorar as informações do servidor DNS do servidor DHCP

nmcli con mod	arquivo ifcfg- *	Efeito
connection.autoconnect yes	ONBOOT=YES	Ativa automaticamente a conexão na inicialização
connection.id eth0	NAME=eth0	O nome desta conexão
connection.interface-name eth0	DEVICE=eth0	A conexão está ligada a esta interface de rede com este nome
802-3-ethernet.mac-address . . .	HWADDR=. . .	A conexão está ligada à interface de rede com este endereço MAC

Breve lista de sintaxe de comandos nmcli

Comando	Propósito
nmcli dev status	Mostra o status do Network Manager de todas as interfaces de rede
nmcli con show	Liste todas as conexões
nmcli con show name	Liste as configurações atuais para o nome da conexão
nmcli con add con-name name ..	Adicionar uma nova conexão chamada nome
nmcli con mod name ..	Modifique o nome da conexão
nmcli con reload	Diga ao networkManager para reler os arquivos de configuração (útil depois de editados manualmente)
nmcli con up name	Ative o nome da conexão
nmcli dev dis dev	Desative e desconecte a conexão atual na interface de rede dev
nmcli con del name	Exclua o nome da conexão e seu arquivo de configuração

Comando `nmcli`

Um bom ponto de partida seria verificar nossos dispositivos:

```
[root@localhost joatham]# nmcli device status
DEVICE      TYPE      STATE                   CONNECTION
enp0s8      ethernet  conectado               enp0s8
enp0s3      ethernet  conectado               enp0s3
virbr0      bridge    connected (externally)  virbr0
lo          loopback  não gerenciável         --
virbr0-nic  tun       não gerenciável         --
```

Como podemos ver na primeira coluna, esta é uma lista de nossos dispositivos de rede. Temos uma placa de rede com nome `enp0s3`. Em sua máquina, você poderá ver outros nomes.

A nomenclatura depende do tipo de placa de rede (se estiver integrada, placa pci, etc). Na última coluna, vemos nossos arquivos de configuração que são usados por nossos dispositivos para se conectar à rede.

É simples entender que nossos dispositivos por si próprios nada podem fazer. Eles precisam que façamos um arquivo de configuração para dizer-lhes como obter conectividade de rede. Chamamos esses arquivos também de "perfis de conexão" . Nós os encontramos no diretório `/etc/sysconfig/network-scripts`.

```
[root@localhost joatham]# cd /etc/sysconfig/network-scripts/
[root@localhost network-scripts]# ls
ifcfg-enp0s3  ifcfg-enp0s3-1  ifcfg-enp0s8
```

Como você pode ver aqui, os arquivos com o nome começando com `ifcfg-`(configuração da interface) são perfis de conexão. Quando criamos uma nova conexão ou modificamos uma existente com `nmcli` ou `nmtui`, os resultados são salvos aqui como perfis de conexão.

```
[root@localhost network-scripts]# cat ifcfg-enp0s3
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp0s3
UUID=7e2d3b19-01af-47f5-baf1-2e032f00e719
```

```
DEVICE=enp0s3
ONBOOT=no
```

Percebemos que algumas propriedades têm valores diferentes e outras não existem se não for necessário. Vamos dar uma olhada rápida no mais importante deles:

- **TYPE** - Temos o tipo `Ethernet` aqui. Poderíamos ter `wi-fi`, `team`, `bond` e outros.
- **DEVICE** - O nome do dispositivo de rede associado a este perfil.
- **BOOTPROTO** - Se tiver o valor `dhcp`, então nosso perfil de conexão recebe IP dinâmico do servidor dhcp. Se tiver o valor `none`, então não leva nenhum IP dinâmico e provavelmente quando atribuímos um IP estático.
- **IPADDR** - É o IP estático que atribuímos ao nosso perfil.
- **PREFIX** - A máscara de sub-rede. Um valor de 24 significa 255.255.255.0. Você pode entender melhor a máscara de sub-rede se anotar seu formato binário. Por exemplo, valores de 16, 24, 26 significam que os primeiros 16, 24 ou 26 bits respectivamente são 1 e o resto 0, definindo exatamente qual é o endereço de rede e qual é a faixa de ip que pode ser atribuída.
- **GATEWAY** - O IP do gateway.
- **DNS1**, **DNS2** - Dois servidores DNS que deseja usar.
- **ONBOOT** - Se tiver o valor `yes` significa que, na inicialização, nosso computador vai ler este perfil e tentar atribuí-lo ao seu dispositivo.

Hands On

- Verifique se o *NetworkManager* está em execução Você pode usar o comando abaixo para verificar se o *NetworkManager* está rodando ou não.

```
[root@localhost network-scripts]# nmcli -t -f RUNNING general
running
```

Para obter um status geral:

```
[root@localhost network-scripts]# nmcli general
STATE      CONNECTIVITY  WIFI-HW     WIFI        WWAN-HW     WWAN
conectado  completa      habilitado  habilitado  habilitado  habilitado
```

- Listar todos os dispositivos disponíveis

Para ver e listar todos os dispositivos disponíveis em seu sistema Linux

```
[root@localhost network-scripts]# nmcli device status
DEVICE      TYPE      STATE                   CONNECTION
enp0s8      ethernet  conectado               enp0s8
enp0s3      ethernet  conectado               enp0s3
virbr0      bridge    connected (externally)  virbr0
lo          loopback  não gerenciável         --
virbr0-nic  tun       não gerenciável         --
```

- Listar todas as conexões disponíveis

Para listar todas as conexões disponíveis:

```
[root@localhost network-scripts]# nmcli connection show enp0s8
connection.id:                          enp0s8
connection.uuid:                        cf11210c-1e71-46c4-aaa1-0c9c3df73b83
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              enp0s8
connection.autoconnect:                 sim
connection.autoconnect-priority:        0
```

- Altere o nome do host usando `nmcli`

Idealmente, você pode alterar o nome do host usando o comando `hostnamectl`, mas também pode atualizar o nome do host usando `nmcli`.

Para obter o nome do host atual:

```
[root@localhost network-scripts]# nmcli general hostname
localhost.localdomain
```

Atualizar o nome do host:

```
[root@localhost network-scripts]# nmcli general hostname rhel8.4linux.com.br
[root@localhost network-scripts]# nmcli general hostname
rhel8.4linux.com.br
[root@localhost network-scripts]# hostname
rhel8.4linux.com.br
```

- Crie uma nova conexão `Ethernet` e atribua um endereço IP estático:

Neste exemplo, `nmcli` configura a interface `enp0s8` estaticamente, usando o endereço `IPv4` e o prefixo de rede `192.168.15.15/24` e o gateway padrão `192.168.15.1`, mas ainda se conecta automaticamente na inicialização e salva sua configuração no arquivo em `/etc/sysconfig/network-scripts/ifcfg-enp0s8` .

```
[root@localhost joatham]# nmcli con add con-name enp0s8 type ethernet ifname enp0s8 ipv4.
     method manual ipv4.addresses 192.168.15.15/24 ipv4.gateway 192.168.15.1
A conexão “”enp0s8 (cf11210c-1e71-46c4-aaa1-0c9c3df73b83) foi adicionada com sucesso.
```

- Crie uma nova conexão Ethernet e atribua um endereço IP DHCP

O comando a seguir adicionará uma nova conexão para a interface `enp0s3`, que obterá informações de rede IPv4 usando `DHCP` e se conectará automaticamente na inicialização. A configuração será salva `/etc/sysconfig/network-scripts/ifcfg-enp0s3` porque o `con-name` é `enp0s3`

```
[root@localhost joatham]# nmcli con add con-name enp0s3 type ethernet ifname enp0s3 ipv4.
     method auto
Aviso: Há outra conexão com o nome “”enp0s3. Referencie a conexão por seu uuid “6f5a9e12-
     a2f9-4b22-9cd3-2043516”acdb9
A conexão “”enp0s3 (6f5a9e12-a2f9-4b22-9cd3-2043516acdb9) foi adicionada com sucesso.
```

```
[root@localhost network-scripts]# egrep BOOTPROTO /etc/sysconfig/network-scripts/ifcfg-
     enp0s3
BOOTPROTO=dhcp
```

- Recarregue a conexão usando `nmcli` (reiniciar)

Recarregue todos os arquivos de conexão do disco. O `NetworkManager` não monitora alterações nos arquivos de conexão por padrão. Portanto, você precisa usar este comando para dizer ao `NetworkManager` para reler os perfis de conexão do disco quando uma alteração foi feita neles.

```
[root@localhost network-scripts]# nmcli con reload
```

- Adicionar/editar interativamente uma conexão

Você pode usar `nmcli con edit` para editar uma conexão existente ou adicionar uma nova, usando um editor interativo. No exemplo abaixo, vamos editar o endereço IP de `enp0s8`

```
[root@localhost network-scripts]# nmcli con edit enp0s8

===| editor interativo de conexões do nmcli |===

Editando conexão existente “802-3-”ethernet: “”enp0s8

Digite “”help ou “”? para comandos disponíveis.
Digite “”print para mostrar todas as propriedades de conexão.
Digite “describe [<definição>.<propriedade”>] para exibir descrição detalhada da
     propriedade.

Você pode editar as seguintes configurações: connection, 802-3-ethernet (ethernet), 802-1x,
      dcb, sriov, ethtool, match, ipv4, ipv6, hostname, tc, proxy
nmcli>
```

Agora verifique suas mudanças no arquivo de configuração de `enp0s8`.

```
[root@localhost network-scripts]# egrep IPADDR /etc/sysconfig/network-scripts/ifcfg-enp0s8
IPADDR=192.168.15.16
```

- Altere a conexão Ethernet `BOOTPROTO` de Estático para `DHCP`

Similarmente, para alterar a conexão Ethernet `BOOTPROTO` de estática para `DHCP` usando `nmcli`, devemos modificar a diretiva `ipv4.method` para usar `auto`.

```
[root@localhost network-scripts]# nmcli connection modify enp0s8 ipv4.method auto

[root@localhost network-scripts]# egrep 'BOOTPROTO|IPADDR|PREFIX|GATEWAY' /etc/sysconfig/
     network-scripts/ifcfg-enp0s8
BOOTPROTO=dhcp
IPADDR=192.168.15.15
PREFIX=24
GATEWAY=192.168.15.1

[root@localhost network-scripts]# ip addr show dev enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
      qlen 1000
    link/ether 08:00:27:88:22:2c brd ff:ff:ff:ff:ff:ff
    inet 192.168.15.15/24 brd 192.168.15.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
    inet6 2804:1b2:b880:63:f23f:fe06:1bf1:9f3/64 scope global dynamic noprefixroute
       valid_lft 43163sec preferred_lft 43163sec
    inet6 fe80::9952:b2b9:ba54:8d6a/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

- Altere a diretiva `ONBOOT` usando `nmcli`

Por padrão, `ONBOOT` é setado como `YES` no arquivo de configuração da interface. Portanto, para

desativar o ONBOOT, devemos modificar a diretiva connection.autoconnect usando nmcli.

Verifique o valor ONBOOT antes de alterar esta diretiva.

```
[root@localhost network-scripts]# egrep 'ONBOOT' /etc/sysconfig/network-scripts/ifcfg-
     enp0s8
ONBOOT=yes
[root@localhost network-scripts]# nmcli connection modify enp0s8 connection.autoconnect no
[root@localhost network-scripts]# egrep 'ONBOOT' /etc/sysconfig/network-scripts/ifcfg-
     enp0s8
ONBOOT=no
```

- Desative o endereço IPv6 para conexão Ethernet (IPV6INIT)

Por padrão, os tipos de conexão IPv4 e IPv6 (IPV6INIT) são habilitados para qualquer tipo de conexão Ethernet. Para usar apenas IPv4 e desabilitar IPv6 usando nmcli.

Verifique o status existente do tipo de conexão IPv6 para enp0s8.

```
[root@localhost network-scripts]# egrep 'IPV6INIT' /etc/sysconfig/network-scripts/ifcfg-
     enp0s8
IPV6INIT=yes

[root@localhost network-scripts]# nmcli connection modify enp0s8 ipv6.method ignore

[root@localhost network-scripts]# egrep 'IPV6INIT' /etc/sysconfig/network-scripts/ifcfg-
     enp0s8
IPV6INIT=no
```

- Ative/Desative uma conexão.

Desative uma conexão de um dispositivo sem impedir que o dispositivo seja mais auto ativado usando nmcli con down <ifname>.

Várias conexões podem ser passadas para o comando.

```
[root@localhost network-scripts]# nmcli connection down enp0s8
Conexão “”enp0s8 desativada com sucesso (caminho D-Bus ativo: /org/freedesktop/
     NetworkManager/ActiveConnection/3)

[root@localhost network-scripts]# nmcli connection show --active
NAME    UUID                                  TYPE      DEVICE
enp0s3  6f5a9e12-a2f9-4b22-9cd3-2043516acdb9  ethernet  enp0s3
virbr0  73d92c5d-694b-42a8-af7c-721974e271f6  bridge    virbr0
```

```
[root@localhost network-scripts]# nmcli connection up enp0s8
Conexão ativada com sucesso (caminho D-Bus ativo: /org/freedesktop/NetworkManager/
     ActiveConnection/5)

[root@localhost network-scripts]# nmcli connection show --active
NAME    UUID                                  TYPE      DEVICE
enp0s3  6f5a9e12-a2f9-4b22-9cd3-2043516acdb9  ethernet  enp0s3
enp0s8  cf11210c-1e71-46c4-aaa1-0c9c3df73b83  ethernet  enp0s8
virbr0  73d92c5d-694b-42a8-af7c-721974e271f6  bridge    virbr0
```

24

Configurar a resolução de nome do host

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar rede básica**
  - Configurar endereços IPv4 e IPv6
  - **Configurar resolução de nome de host**
  - Configurar os serviços de rede para iniciar automaticamente na inicialização
  - Restringir o acesso à rede usando firewall-cmd/firewall

Introdução

Outra configuração de rede imprescindível de ser feita em qualquer máquina é a alteração do seu Hostname e configurações de DNS, quer sejam locais ou Remotas. No RHEL, existem 3 tipos de hostnames: `static`, `pretty` e `transient`.

- *`Static`* - O hostname Static (estático) é o mais tradicional, sendo aquele escolhido pelo utilizador e normalmente encontra-se armazenado no arquivo `/etc/hostname` (Note que este arquivo não deve ser editado manualmente), este tipo de hostname usa por default

o valor de localhost.

- *Transient* - O Transient (transitório) é um hostname dinâmico gerido pelo kernel. É inicializado baseado no hostname estático, contudo, pode ser alterado por outros serviços como o DCHP ou mDNS, durante o runtime da máquina.
- *Pretty* - O Pretty hostname é um nome livre de base UTF-8 que serve para apresentação ao usuário.

Para obtermos os hostnames da máquina em que estamos trabalhando, usamos o comando `hostnamectl`:

```
[root@rhel8 joatham]# hostnamectl
   Static hostname: rhel8.4linux.com.br
         Icon name: computer-vm
           Chassis: vm
        Machine ID: b002bcbcf6d54646b9265924d6266e20
           Boot ID: 75ffba272f4c485aa16dfcb4831fa593
    Virtualization: oracle
  Operating System: Red Hat Enterprise Linux 8.4 (Ootpa)
       CPE OS Name: cpe:/o:redhat:enterprise_linux:8.4:GA
            Kernel: Linux 4.18.0-305.19.1.el8_4.x86_64
      Architecture: x86-64
```

Contudo, caso seja pretendido apenas obter o valor existente em `/etc/hostname`, podemos usar apenas:

```
[root@rhel8 joatham]# hostname
rhel8.4linux.com.br
```

Caso o Administrador pretenda alterar o `hostname` da máquina, deverá usar novamente o `hostnamectl`, mas definindo a opção `set-hostname`:

```
# hostnamectl set-hostname [Nome]
```

Tendo um Hostname definido é essencial garantirmos que a nossa máquina seja reconhecida numa rede pelo seu nome e que esta conheça também as outras máquinas na rede pelos seus respectivos nomes.

Para conseguirmos isso, temos primeiro que perceber como é que o nosso sistema está preparado para agir em conformidade, ou seja, que métodos utiliza para fazer a Resolução de Nomes. Para validarmos isso, devemos consultar o arquivo `/etc/nsswitch.conf`, que conterá uma linha semelhante a esta:

```
hosts: files dns
```

Ou seja, para identificação de Hosts, esta máquina primeiro vai recorrer aos arquivos (resolução estática, através do `/etc/hosts`), e depois recorrerá ao serviço de DNS (resolução dinâmica).

Quando um sistema recorre ao `/etc/hosts` ele vai procurar por linhas que respeitem a seguinte sintaxe:

```
[IP_Local] [Hostname+Dominio] [Hostname]
```

Por exemplo:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
```

Quando um sistema recorre ao serviço de `DNS`, este consulta o arquivo `/etc/resolv.conf` para saber quais os servidores na rede é que estão responsáveis por fazer a resolução de domínios. Um exemplo deste arquivo seria:

```
; generated by /usr/sbin/dhclient-script
search ifto.local. 4linux.com.br
nameserver 10.11.0.5
nameserver 10.0.30.17
nameserver 8.8.8.8
nameserver 1.1.1.1
```

Podemos ter num sistema até 3 *NameServer's* diferentes. Eles funcionam em cadeia, para caso o de cima não funcione ou não consiga identificar - assim, se passa para o seguinte, repetindo o processo até que haja uma resposta ou nenhuma resposta por todos os *NameServer's*.

Note ainda que, tal como especificado no arquivo, este foi gerado automaticamente pelo *NetworkManager* - lembrando que não deve ser editado manualmente. Antes, é preciso tomar o recurso do comando `nmcli` de forma a especificar alterações nas configurações:

```
# nmcli con mod [Placa_Rede] +ipv4.dns [NameServer]
```

Pode especificar `-ipv4.dns`para remover ou `ipv4.dns`para substituir um valor existente.# Con-

figurar serviços de rede para iniciar automaticamente na inicialização

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar rede básica**
  - Configurar endereços IPv4 e IPv6
  - Configurar resolução de nome de host
  - **Configurar os serviços de rede para iniciar automaticamente na inicialização**
  - Restringir o acesso à rede usando firewall-cmd/firewall

Hands On

- Verificando se o serviço de rede está ativado:

```
[root@rhel8 joatham]# systemctl is-active NetworkManager
active
[root@rhel8 joatham]# systemctl enable NetworkManager
```

- Instalando Servidor Web e colocando-o na inicialização

```
[root@rhel8 joatham]# dnf install httpd
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 20:25:46 atrás em dom 10 out 2021
     20:30:15 -03.
Dependências resolvidas.
================================================================================
 Pacote              Arq.   Versão        Repositório                       Tamanho
================================================================================
Instalando:
 httpd               x86_64 2.4.37-39.module+el8.4.0+9658+b87b2deb
                                          rhel-8-for-x86_64-appstream-rpms 1.4 M
Instalando dependências:
 apr                 x86_64 1.6.3-11.el8  rhel-8-for-x86_64-appstream-rpms 125 k
 apr-util            x86_64 1.6.1-6.el8   rhel-8-for-x86_64-appstream-rpms 105 k
 httpd-filesystem    noarch 2.4.37-39.module+el8.4.0+9658+b87b2deb
                                          rhel-8-for-x86_64-appstream-rpms  38 k
 httpd-tools         x86_64 2.4.37-39.module+el8.4.0+9658+b87b2deb
                                          rhel-8-for-x86_64-appstream-rpms 106 k
 mod_http2           x86_64 1.15.7-3.module+el8.4.0+8625+d397f3da
                                          rhel-8-for-x86_64-appstream-rpms 154 k
 redhat-logos-httpd noarch 84.4-1.el8     rhel-8-for-x86_64-baseos-rpms     29 k
Instalando dependências fracas:
 apr-util-bdb        x86_64 1.6.1-6.el8   rhel-8-for-x86_64-appstream-rpms  25 k
```

```
 apr-util-openssl   x86_64 1.6.1-6.el8   rhel-8-for-x86_64-appstream-rpms  27 k
Ativando Fluxos de Módulos:
 httpd                    2.4

Resumo da transação
================================================================================
Instalar  9 Pacotes

Tamanho total do download: 2.0 M
Tamanho depois de instalado: 5.4 M
Correto? [s/N]: s
```

```
[root@rhel8 joatham]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/
     systemd/system/httpd.service.
```

- Colocando a interface de rede para inicializar junto com o sistema:

```
[root@rhel8 joatham]# nmcli connection modify enp0s8 connection.autoconnect yes
[root@rhel8 joatham]# egrep 'ONBOOT' /etc/sysconfig/network-scripts/ifcfg-enp0s8
ONBOOT=yes
```

25

Restringir acesso à rede usando firewall-cmd/firewall

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar rede básica**
  - Configurar endereços IPv4 e IPv6
  - Configurar resolução de nome de host
  - Configurar os serviços de rede para iniciar automaticamente na inicialização
  - **Restringir o acesso à rede usando firewall-cmd/firewall**

Trabalhando com Zonas

As zonas representam um conceito para gerenciar o tráfego de entrada de forma mais transparente. Elas são conectadas a interfaces de rede ou atribuídas a um intervalo de endereços de origem. Você gerencia regras de firewall para cada zona de forma independente, o que permite definir configurações de firewall complexas e aplicá-las ao tráfego.

Hands On

- Checando as Zonas existentes no firewall-cmd.

```
[root@rhel8 joatham]# firewall-cmd --get-zones
block dmz drop external home internal libvirt nm-shared public trusted work
```

- Visualizando tudo que pertence a zona `work`.

```
[root@rhel8 joatham]# firewall-cmd --zone=work  --list-all
work
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

- Criando uma zona restrita.

Para usar zonas personalizadas, crie uma nova zona e use-a como uma zona predefinida. Novas zonas exigem a opção `--permanent`, caso contrário, o comando não funciona.

```
[root@rhel8 joatham]# firewall-cmd --new-zone 4linux --permanent
success
```

```
[root@rhel8 joatham]# firewall-cmd --reload
success
```

```
[root@rhel8 joatham]# firewall-cmd --get-zones
4linux block dmz drop external home internal libvirt nm-shared public trusted work
```

Criação de uma nova zona usando um arquivo de configuração

As zonas também podem ser criadas usando um arquivo de configuração de zona. Essa abordagem pode ser útil quando você precisa criar uma nova zona, mas deseja reutilizar as configurações de uma zona diferente e apenas alterá-las um pouco.

Um arquivo `firewalld` de configuração de zona contém as informações, como a descrição da zona, serviços, portas, protocolos, blocos icmp, masquerade, portas de encaminhamento e regras de linguagem rica em um formato de arquivo XML. O nome do arquivo deve estar `zone-name.xml` - observação: o comprimento do nome da zona está atualmente limitado a `17` caracteres. Os arquivos de configuração da região estão localizados nos diretórios `/usr/lib/firewalld/zones/` e `/etc/firewalld/zones/`.

- Adicionando serviços específicos à zona restrita.

```
[root@rhel8 joatham]# firewall-cmd --zone=4linux --add-service=ssh --permanent
success
```

```
[root@rhel8 joatham]# firewall-cmd --reload
success
```

```
[root@rhel8 joatham]# firewall-cmd --zone=4linux --list-all
4linux
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

- Adicionando interfaces específicas a zonas restritas.

```
[root@rhel8 joatham]# firewall-cmd --change-interface=enp0s8 --zone=4linux --permanent
The interface is under control of NetworkManager, setting zone to '4linux'.
success
```

```
[root@rhel8 joatham]# firewall-cmd --zone=4linux --list-all
4linux (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s8
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
[root@rhel8 joatham]# firewall-cmd --get-active-zones
4linux
  interfaces: enp0s8
public
  interfaces: enp0s3
```

- Definindo zona específica como zona padrão.

```
[root@rhel8 joatham]# firewall-cmd --set-default-zone=4linux
success
```

```
[root@rhel8 joatham]# firewall-cmd --reload
success
```

- Adicionando novos serviços à zona padrão (4linux).

```
[root@rhel8 joatham]# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client amqp amqps apcupsd audit bacula bacula-client
      bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph
     ceph-mon cfengine cockpit collectd condor-collector ctdb dhcp dhcpv6 dhcpv6-client
     distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch
     etcd-client etcd-server finger freeipa-4 freeipa-ldap freeipa-ldaps freeipa-
     replication freeipa-trust ftp galera ganglia-client ganglia-master git grafana gre
     high-availability http https imap imaps ipp ipp-client ipsec irc ircs iscsi-target
     isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-
     apiserver ldap ldaps libvirt libvirt-tls lightning-network llmnr managesieve matrix
     mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nfs
      nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-
     vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy
     prometheus proxy-dhcp ptp pulseaudio puppetmaster quassel radius rdp redis redis-
     sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane
      sip sips slp smtp smtp-submission smtps snmp snmptrap spideroak-lansync spotify-sync
```

```
      squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui synergy syslog syslog
      -tls telnet tentacle tftp tftp-client tile38 tinc tor-socks transmission-client upnp-
      client vdsm vnc-server wbem-http wbem-https wsman wsmans xdmcp xmpp-bosh xmpp-client
      xmpp-local xmpp-server zabbix-agent zabbix-server
```

```
[root@rhel8 joatham]# firewall-cmd --add-service=http --permanent
success
```

```
[root@rhel8 joatham]# firewall-cmd --add-service=https --permanent
success
```

```
[root@rhel8 joatham]# firewall-cmd --add-port=8080/tcp --permanent
success
```

```
[root@rhel8 joatham]# firewall-cmd --reload
success
```

```
[root@rhel8 joatham]# firewall-cmd --zone=4linux --list-all
4linux (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: http https ssh
  ports: 8080/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

26

Criar, excluir e modificar contas de usuário locais

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar usuários e grupos**
  - **Criar, excluir e modificar contas de usuários locais**
  - Alterar as senhas e ajustar o vencimento da senha para contas de usuários locais
  - Criar, excluir e modificar grupos locais e associações a grupos
  - Configurar o acesso de superusuário

Introdução

Um usuário pode ser representado como qualquer pessoa que esteja disposta a usar a máquina, usar o computador ou, melhor, o sistema operacional (S.O.).

Agora, o nome que você escolhe para identificar o seu usuário é irrelevante, ou seja, não interessa se o nome dado é real ou um nome fictício. O S.O. simplesmente o aceita, pois associa esse nome sempre a um número identificador, o `UID-User ID`.

Isso é otimo porque, quando vamos criar um usuário, podemos personalizar inúmeros aspectos

da sua conta. Como, por exemplo, a sua data de expiração de senhas, o grupo ao qual ele pertence, onde se localiza, o seu diretório home, entre muitas outras opções. E o melhor: todas elas podem ser consultadas na página `man` dos comandos `useradd` e `usermod`.

Por padrão, para criar qualquer usuário no sistema, é usado o comando `useradd`. Já para alterarmos alguma configuração no usuário que existen, usamos o comando `usermod`. Você pode estar se perguntando... "Mas e se eu quiser remover esse usuário do meu Sistema Operacional?". Muito facil! Basta utilizar o comando `userdel`. Dito isso, vamos ao que interessa:

- Criação simples de um usuário:

```
# useradd [usuario]
```

- Criação de um usuário definindo um UID e GID específicos:

```
# useradd -u [UID] -g [GID] [usuario]
```

> Se liga: se a opção usada for `-g`, o grupo que é definido a seguir será o grupo principal do usuário. Agora se a opção for `-G`, o grupo que for indicado será definido apenas como secundário.

Caso algum usuário precise ser deletado:

```
# userdel [usuario]
```

> Um detalhe importante é que esse comando apaga o usuário, mas mantendo todos as pastas no seu diretório home intactas. Caso queira eliminar tudo referente aquele usuário, opte pela a opção `-r`:

```
# userdel -r [usuario]
```

Comando usermod

O `usermod` é um utilitário de linha de comando que permite modificar as informações de login de um usuário. No **RHCSA**, é muito provável que sejam solicitados aos examinandos para serem criadas e alteradas diferentes configurações das contas dos users, sendo que abaixo ficam algumas possíveis alterações aos usuários:

A sintaxe do comando `usermod` assume a seguinte forma:

```
usermod [options] USER
```

Apenas root ou usuários com acesso sudo podem invocar o `usermod` e modificar uma conta de usuário. Em caso de sucesso, o comando não exibe nenhuma saída!

Adicionar um usuário a um grupo

O caso de uso mais típico do `usermod` é adicionar um usuário a um grupo. Para adicionar um usuário existente a um grupo secundário, use as opções `-a -G` após o nome do grupo e o nome de usuário:

```
usermod -a -G GROUP USER
```

Se você deseja adicionar o usuário a vários grupos de uma vez, especifique os grupos após a opção `-G` separados por ,(vírgulas) sem nenhum espaço em branco intermediário. Por exemplo, para adicionar o usuário joatham ao grupo 4linux, você executaria o seguinte comando:

```
[root@localhost joatham]# usermod -a -G 4linux joatham
```

Sempre use a opção `-a` (anexar) ao adicionar um usuário a um novo grupo. Se você omitir a opção `-a`, o usuário será removido dos grupos não listados após a opção `-G`.

Se o usuário ou grupo não existir, o comando vai avisá-lo.

Alterar o grupo primário do usuário

Para alterar o grupo primário de um usuário, invoque o comando `usermod` com a opção `-g` seguida do nome do grupo e do nome de usuário:

```
[root@localhost joatham]# usermod -g GROUP USER
```

No exemplo a seguir, estamos alterando o grupo principal do usuário joatham para infra:

```
[root@localhost joatham]# usermod -g infra joatham
```

Cada usuário pode pertencer a exatamente um grupo primário, e zero ou mais grupos secundários.

Alteração das informações do usuário

Para alterar as informações do GECOS (nome completo do usuário), execute o comando com a opção `-c` seguida pelo novo comentário e nome de usuário:

```
usermod -c "GECOS Comment" USER
```

Aqui está um exemplo que mostra como adicionar informações adicionais ao usuário joatham:

```
[root@localhost joatham]# usermod -c "Usuario de teste" joatham
```

> Essas informações são armazenadas no arquivo `/etc/passwd`.

Alterando o diretório inicial de um usuário

Na maioria dos sistemas Linux, os diretórios pessoais do usuário são nomeados de acordo com o nome do usuário e criados no diretório `/home`.

Se, por algum motivo, você deseja alterar o diretório inicial do usuário, invoque o comando `usermod` com a opção `-d` seguida do caminho absoluto do novo diretório inicial e o nome do usuário:

```
usermod -d HOME_DIR USER
```

Por padrão, o comando não move o conteúdo do diretório inicial do usuário para o novo. Para mover o conteúdo, use a opção `-m`. Se o novo diretório ainda não existir, ele será criado:

```
usermod -d HOME_DIR -m USER
```

Aqui está um exemplo que mostra como alterar o diretório inicial do usuário `www-data` para `/var/www`:

```
[root@localhost joatham]# usermod -d /var/www www-data
```

Mudando um Shell Padrão do Usuário

O shell padrão é o shell que é executado após o login no sistema. Por padrão, na maioria dos sistemas Linux, o shell padrão é definido como `Bash Shell`. Para alterar o shell padrão do usuário, execute o comando com a opção `-s` seguida do caminho absoluto do shell e o nome do usuário:

```
usermod -s SHELL USER
```

No exemplo abaixo, estamos alterando o shell do usuário para Zsh:

```
[root@localhost joatham]# usermod -s /usr/bin/zsh joatham
```

Em algumas distribuicões, você pode descobrir quais shells estão disponíveis em seu sistema exibindo o conteúdo do arquivo `/etc/shells`.

Alterando um UID de usuário

UID (o identificador do usuário) é um número atribuído a cada usuário. É usado pelo sistema operacional para se referir a um usuário.

Para alterar o UID do usuário, invoque o comando com a opção `-u` seguido do novo UID e o nome do usuário:

```
usermod -u UID USER
```

O exemplo abaixo mostra como alterar o número "UID" para "1050":

```
[root@localhost joatham]# usermod -u 1050 joatham
```

O UID dos arquivos pertencentes ao usuário estão localizados no diretório inicial do usuário, e o arquivo da caixa de correio do usuário será alterado automaticamente. A propriedade de todos os outros arquivos deve ser alterada manualmente.

Alterar um nome de usuário

Embora não seja muito frequente, às vezes você pode querer alterar o nome de um usuário existente. A opção `-l` é usada para alterar o nome de usuário:

```
usermod -l NEW_USER USER
```

No exemplo abaixo, estamos renomeando o usuário `joatham` para `pedro`:

```
[root@localhost joatham]# usermod -l joatham pedro
```

Ao alterar o nome de usuário, você também pode alterar o diretório pessoal do usuário para refletir o novo nome de usuário.

Definir uma data de expiração do usuário

A data de expiração é a data em que a conta do usuário será desabilitada. Para definir a data de expiração do usuário, use a opção `-e`:

```
sudo usermod -e DATE USER
```

A data de validade deve ser definida usando o formato `YYYY-MM-DD`.

Por exemplo, para desabilitar o usuário joatham em 2022-02-21, você executaria o seguinte comando:

```
[root@localhost joatham]# usermod -e "2022-02-21" joatham
```

Para desativar a expiração de uma conta, defina uma data de expiração vazia:

```
[root@localhost joatham]# usermod -e "" joatham
```

Use o comando `chage -l` para ver a data de validade do usuário:

```
[root@localhost joatham]# chage -l joatham
Last password change                    : Jul 24, 2018
Password expires                        : never
Password inactive                       : never
Account expires                         : never
Minimum number of days between password change      : 0
Maximum number of days between password change      : 99999
Number of days of warning before password expires   : 7
```

A data de expiração é armazenada no arquivo `/etc/shadow`.

Bloqueio e desbloqueio de uma conta de usuário

A opção `-L` permite que você bloqueie uma conta de usuário:

```
usermod -L USER
```

Os comandos vão inserir um ponto de exclamação (`!`) na frente da senha criptografada. Quando o campo de senha no arquivo `/etc/shadow` contém um ponto de exclamação, o usuário não poderá fazer login no sistema usando autenticação de senha. Outros métodos de login, como autenticação baseada em chave ou mudança para o usuário ainda são permitidos. Se você deseja bloquear a conta e desabilitar todos os métodos de login, você também precisa definir a data de expiração para 1.

Os exemplos a seguir mostram como bloquear o usuário joatham:

```
[root@localhost joatham]# usermod -L joatham
```

```
[root@localhost joatham]# usermod -L -e 1 joatham
```

Para desbloquear um usuário, execute `usermod` com a opção `-U`: `shell usermod -U USER`# Alterar

senhas e ajustar tempo de senha para contas de usuário locais ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar usuários e grupos**
  - Criar, excluir e modificar contas de usuários locais
  - **Alterar as senhas e ajustar o vencimento da senha para contas de usuários locais**
  - Criar, excluir e modificar grupos locais e associações a grupos
  - Configurar o acesso de superusuário

Introdução

Toda vez que criamos um usuario novo temos a possibilidade de definir a senha que ele terá por padrão. Essa tarefa pode ser definida facilmente através da opção `-p` do comando `useradd`. Inclusive até a data que a conta deverá expirar, sendo que para isso basta passar a opção `-e` do comando `useradd`.

Partindo deste pressuposto, vamos tirar a prova criando um `usuario`, com a senha `4linux123`. Além disso, vamos definir que este `usuario` terá a sua conta expirada na data 2022-08-02. Para isso, usamos o seguinte comando:

```
[root@localhost joatham]# useradd -p 4linux123 -e 2022-08-02 Usuario
```

Contudo, ambas as opções indicadas não são obrigatórias. É possível alterar estas configurações mesmo após criação de um determinado usuário.

Por exemplo, vamos imaginar que criamos o nosso `Usuario` e, após a sua criação, pretendíamos mudar a sua senha. Para isso, bastaria o seguinte comando:

```
# passwd [Utilizador]
```

Vale lembrar que uma vez feito login pelo próprio usuário, ele pode utilizar o comando `passwd` novamente de forma a alterar mais uma vez a sua senha.

Tendo já então definido uma senha para o nosso utilizador, usando um dos métodos indicados acima, podemos agora ainda definir períodos de expiração de senha e de conta, de forma a forçar o utilizador a alterar, com alguma regularidade estipulada, a sua senha. Para fazermos

estas alterações, podemos tomar conhecimento do comando `chage`:

O comando chage

O comando chage é usado para modificar as informações de expiração da senha do usuário. Ele permite que você visualize as informações de vencimento da conta do usuário, altere o número de dias entre as alterações de senha e a data da última alteração de senha.

Depois de definir a expiração e envelhecimento da senha, essas informações são usadas pelo sistema para determinar quando um usuário deve alterar sua senha. Normalmente, as empresas ou organizações têm certas políticas de segurança que exigem que os usuários alterem as senhas regularmente: essa pode ser uma maneira simples de aplicar essas políticas.

Para visualizar as informações de vencimento da conta do usuário, use a flag `-l`.

```
[root@localhost joatham]# chage -l joatham
```

Para definir a data ou o número de dias (desde 1 de janeiro de 1970), quando a senha foi alterada pela última vez, use a flag `-d`.

```
[root@localhost joatham]# chage -d 2021-02-11 joatham
```

Em seguida, você também pode definir a data ou o número de dias (desde 1ž de janeiro de 1970) em que a conta do usuário não estará mais acessível, usando a opção `-E`, conforme mostrado no comando a seguir.

Nesse caso, uma vez que a conta de um usuário é bloqueada, ele deve entrar em contato com o administrador do sistema antes de poder usar novamente.

```
[root@localhost joatham]# chage -E 2021-02-16 joatham
```

Além disso, com a opção `-W`, é possivel definir o número de dias de aviso antes que uma alteração de senha seja necessária. Considerando o comando abaixo, o usuário joatham será avisado 10 dias antes do vencimento de sua senha.

```
[root@localhost joatham]# chage -W 10 joatham
```

Você também pode definir o número de dias de inatividade após a expiração de uma senha antes que a conta seja bloqueada. Este exemplo significa que depois que a senha do usuário joatham expirar, sua conta ficará inativa por 2 dias antes de ser bloqueada.

Quando a conta se torna inativa, é necessário entrar em contato com o administrador do sistema antes de poder usar novamente.

```
[root@localhost joatham]# chage -I 2 joatham
```

Depois de tudo isso dito, ainda te aconselhamos fortemente a consultar a página de manual do `chage`

```
[root@localhost joatham]# man chage
```

27

Criar, excluir e modificar grupos locais e membros de grupos

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar usuários e grupos**
  - Criar, excluir e modificar contas de usuários locais
  - Alterar as senhas e ajustar o vencimento da senha para contas de usuários locais
  - **Criar, excluir e modificar grupos locais e associações a grupos**
  - Configurar o acesso de superusuário

Introdução

No Linux, os grupos são usados para organizar e administrar contas de usuários. O objetivo principal dos grupos é definir um conjunto de privilégios, como permissão de leitura, gravação ou execução para um determinado recurso, que pode ser compartilhado entre os usuários do grupo.

Nesse contexto, vamos começar falando sobre como criar novos grupos no Linux usando o comando `groupadd`. Confira!

Sintaxe de Comando `groupadd`

A sintaxe geral do comando `groupadd` é a seguinte:

```
groupadd [OPTIONS] GROUPNAME
```

Apenas o root ou um usuário com privilégios sudo pode criar novos grupos.

Quando chamado, `groupadd` cria um novo grupo usando as opções especificadas na linha de comando mais os valores padrão especificados no arquivo `/etc/login.defs`.

Criação de um grupo no Linux

Para criar um novo tipo de grupo, execute o comando `groupadd` seguido pelo novo nome do grupo. Por exemplo, para criar um novo grupo chamado `meugrupo`, você executaria:

```
[root@localhost joatham]# groupadd meugrupo
```

O comando adiciona uma entrada para o novo grupo aos arquivos `/etc/group` e `/etc/gshadow`.

Depois que o grupo for criado, você pode começar a adicionar usuários ao grupo.

Se o grupo com o mesmo nome já existir, o sistema imprimirá uma mensagem de erro como a seguinte:

```
groupadd: group 'meugrupo' already exists
```

Para suprimir a mensagem de erro se o grupo existir, além de fazer com que o comando saia com sucesso, use a opção `-f`(–force):

```
[root@localhost joatham]# groupadd -f meugrupo
```

Criando um Grupo com GID Específico

No Linux e em sistemas operacionais semelhantes ao Unix, os grupos são identificados por seu nome e um `GID` exclusivo (um número inteiro positivo). Por padrão, quando um novo grupo é criado, o sistema atribui o próximo `GID` disponível do intervalo de IDs de grupo especificado no arquivo `login.defs`.

Use a opção `-g`(`--gid`) para criar um grupo com um `GID` específico.

Por exemplo, para criar um grupo nomeado `meugrupo` com `GID` de `1010`, você digitaria:

```
[root@localhost joatham]# groupadd -g 1010 meugrupo
```

Você pode verificar o `GID` do grupo, listando todos os grupos e filtrando o resultado com grep:

```
[root@localhost joatham]# getent group | grep meugrupo
```

Se já existir um grupo com o `GID` fornecido, aparecerá o seguinte erro:

```
groupadd: GID '1010' already exists
```

Quando usado com a opção `-o`(–non-unique), o comando `groupadd` permite que você crie um grupo com `GID` não exclusivo:

```
[root@localhost joatham]# groupadd -o -g 1010 meugrupo
```

Criando um Grupo de Sistema

Não há diferença técnica real entre o sistema e os grupos regulares (normais). Normalmente, os grupos do sistema são usados para alguns fins especiais de operação do sistema, como criar backups ou fazer manutenção do sistema.

Os `GIDs` dos grupos do sistema são escolhidos a partir do intervalo de `UDs` do grupo do sistema especificado no arquivo `login.defs`, que é diferente do intervalo usado para grupos regulares.

Use a opção `-r`(–system) para criar um grupo de sistema. Por exemplo, para criar um novo grupo de sistema denominado `meugrupodosistema`, você executaria:

```
[root@localhost joatham]# groupadd -r meugrupodosistema
```

Substituindo os valores padrão `/etc/login.defs`

A opção `-K`(–key) seguida de `KEY=VAL` permite que você substitua os valores padrão especificados no arquivo `/etc/login.defs`.

Basicamente, tudo o que você pode substituir são os valores máximo e mínimo dos IDs de grupo normal e do sistema para seleção automática de `GID` ao criar um novo grupo.

Digamos que você queira criar um novo grupo com `GID` no intervalo entre `1200` e `1500`. Para fazer isso, especifique os valores mínimo/máximo, conforme mostrado abaixo:

```
[root@localhost joatham]# groupadd -K GID_MIN=1200 -K GID_MAX=1500 meugrupo
```

Criando um Grupo de Sistema com Senha

Adicionar uma senha a um grupo não tem uso prático e pode causar um problema de segurança, pois mais de um usuário precisará saber a senha. No entanto, se ainda sim tiver interessado em saber, aqui vai:

A opção `-p`(–password) seguida pela senha permite que você defina uma senha para o novo grupo:

```
[root@localhost joatham]# groupadd -p grouppassword meugrupo
```

Como vimos, grupos podem ser criados usando o comando `groupadd`. Mas se um desses grupos não for mais necessário e tiver que ser removido do sistema?

Sintaxe do comando `groupdel`

A sintaxe geral do comando `groupdel` é a seguinte:

```
groupdel [OPTIONS] GROUPNAME
```

> Apenas o root ou um usuário com privilégios sudo pode remover grupos.

> Não é possível remover o grupo principal de um usuário existente sem remover o usuário primeiro.

O comando `groupdel` aceita apenas algumas opções que raramente são usadas.

Excluindo um Grupo no Linux

Para excluir (remover) um determinado grupo do sistema, invoque o comando `groupdel` seguido do nome do grupo.

Por exemplo, para remover um grupo chamado `meugrupo`, você executaria:

```
[root@localhost joatham]# groupdel meugrupo
```

O comando acima remove a entrada do grupo dos arquivos `/etc/group` e `/etc/gshadow`.

Em caso de sucesso, o comando groupdel não imprime nenhuma saída.

Você pode verificar se o grupo foi removido listando todos os grupos a partir do seguinte comando:

```
[root@localhost joatham]# getent group | grep meugrupo
```

Se o grupo que você deseja remover não existir, o sistema imprimirá uma mensagem de erro como a seguinte:

```
groupdel: group 'mygroup' does not exist
```

Tendo já os nossos grupos criados, basta agora preenchermos eles com diferentes usuários.

Para isso, podemos usar o comando `usermod` ou, ainda, um método mais eficaz: o comando `gpasswd`.

Por exemplo, para adicionar o `Usuario` ao `Grupo`:

```
[root@localhost joatham]# gpasswd -a usuario grupo
```

Para remover o `usuario` do `grupo`:

```
[root@localhost joatham]# gpasswd -d usuario grupo
```

Finalmente, um método prático para ver todos os usuários que se encontram em algum determinado `Grupo`. Use o comando `groupmems`:

`shell [root@localhost joatham]# groupmems -g grupo -l`# Configurar acesso de superusuário ## Pontos de estudo para o exame Os candidatos ao exame `RHCSA` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar usuários e grupos**
  - Criar, excluir e modificar contas de usuários locais
  - Alterar as senhas e ajustar o vencimento da senha para contas de usuários locais
  - Criar, excluir e modificar grupos locais e associações a grupos
  - **Configurar o acesso de superusuário**

Introdução

`sudo` é um utilitário de linha de comando projetado para permitir que usuários confiáveis executem comandos como outro usuário, por padrão o usuário `root`.

Geralmente, você tem duas opções para conceder acesso `sudo` a um usuário. O primeiro é adicionando ele ao arquivo `sudoers`, o qual contém informações que definem quais usuários e grupos têm privilégios `sudo`, bem como o nível dos privilégios.

A segunda opção é adicionar o usuário ao grupo `sudo` definido no arquivo `sudoers`. Por padrão, em distribuições baseadas em RedHat, como CentOS e Fedora, os membros do grupo `wheel` recebem privilégios de sudo.

Adicionando usuário ao grupo wheel

Assim, a maneira mais fácil de conceder privilégios `sudo` a um usuário no `RHEL` é adicionar o usuário ao grupo `wheel`.

Estamos assumindo que o usuário já existe. No entanto, se você deseja criar um novo usuário, consulte as aulas passadas.

Para adicionar o usuário ao grupo, execute o comando abaixo como `root` ou outro usuário `sudo`. Altere o "nome de usuário" com o nome ao qual deseja conceder permissões.

```
usermod -aG wheel username
```

Conceder acesso sudo usando este método é suficiente para a maioria dos casos de uso. Para testar o acesso `sudo`, execute o comando `whoami`:

```
sudo whoami
```

Nessa parte, você será solicitado a inserir a senha. Se o usuário tiver acesso sudo, o comando imprimirá `root`:

```
root
```

> Se você receber um erro dizendo "o usuário não está no arquivo sudoers", significa que o usuário não tem privilégios sudo.

Adicionando usuário ao arquivo sudoers

Os privilégios de `sudo` dos usuários e grupos são configurados no arquivo `/etc/sudoers`. Adicionar o usuário a este arquivo permite conceder acesso personalizado aos comandos e configurar políticas de segurança personalizadas para o usuário.

Você pode configurar o acesso `sudo` do usuário modificando o arquivo `sudoers` ou criando um novo arquivo de configuração no diretório `/etc/sudoers.d`. Os arquivos dentro desse diretório são incluídos no arquivo `sudoers`.

Para editar o arquivo `/etc/sudoers`, use o comando `visudo`. Este comando verifica se há erros de sintaxe no arquivo ao salvá-lo. Se houver algum erro, o arquivo não será salvo. Se você abrir o arquivo com um editor de texto, um erro de sintaxe pode resultar na perda do acesso ao `sudo`. Normalmente, `visudo` usa o `vim` para abrir o `/etc/sudoers`. Se você não tem experiência com o `vim` e deseja editar o arquivo com o editor `nano`, terá de configurar sua variável desta forma:

```
EDITOR=nano visudo
```

Digamos que você queira permitir que o usuário execute comandos sudo sem que seja solicitada uma senha. Para isso, abra o arquivo `/etc/sudoers`:

```
[root@localhost joatham]# visudo
```

Role para baixo até o final do arquivo e adicione a seguinte linha:

```
username  ALL=(ALL) NOPASSWD:ALL
```

Então, salve o arquivo e feche o editor. Não se esqueça de alterar o "nome de usuário" com o nome de usuário ao qual deseja conceder acesso.

Outro exemplo comum é permitir que o usuário execute apenas comandos específicos via `sudo`. Por exemplo, para permitir apenas os comandos `du` e `ping`, você usaria:

```
username ALL=(ALL) NOPASSWD:/usr/bin/du,/usr/bin/ping
```

Em vez de editar o arquivo sudoers, você pode fazer o mesmo criando um novo arquivo com as regras de autorização no diretório `/etc/sudoers.d`. Adicione a mesma regra que você adicionaria ao arquivo `sudoers`:

```
[root@localhost joatham]# echo "username  ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d
     /username
```

Essa abordagem torna o gerenciamento dos privilégios do sudo mais sustentável. O nome do

arquivo não é importante! No entanto, é uma prática comum que o nome do arquivo seja igual ao nome do usuário.# Defina as configurações do firewall usando firewall-cmd/firewalld ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar segurança**
  - **Defina as configurações do firewall usando firewall-cmd/firewalld**
  - Crie e use listas de controle de acesso a arquivos
  - Configure a autenticação baseada em chave para SSH
  - Defina modos de aplicação e permissivos para SELinux
  - Liste e identifique o contexto de processos e arquivos do SELinux
  - Restaure contextos de arquivo padrão
  - Use as configurações booleanas para modificar as configurações do SELinux do sistema
  - Diagnostique e resolva violações de rotina da política do SELinux

Introdução

Firewalld é uma solução de gerenciamento de firewall disponível para muitas distribuições Linux. Atua como um front-end para o sistema de filtragem de pacotes iptables fornecido pelo kernel Linux. Nesta aula, abordaremos como configurar um firewall para o seu servidor e mostraremos os fundamentos do gerenciamento do firewall com a ferramenta administrativa `firewall-cmd`.

Conceitos básicos em Firewalld

Antes de começarmos a falar sobre como realmente usar o utilitário `firewall-cmd` para gerenciar a configuração do firewall, devemos nos familiarizar com alguns conceitos básicos que a ferramenta apresenta.

Zonas

O daemon `firewalld` gerencia grupos de regras usando entidades chamadas "zonas". As zonas são basicamente conjuntos de regras que determinam qual tráfego deve ser permitido, dependendo do nível de confiança que você tem nas redes às quais seu computador está conectado. As interfaces de rede são atribuídas a uma zona para ditar o comportamento que o firewall deve permitir.

Para computadores que podem se mover entre redes com frequência (como laptops), esse tipo de flexibilidade fornece um bom método de alterar suas regras, dependendo de seu ambiente. Você pode ter regras rígidas que proíbem a maior parte do tráfego ao operar em uma rede WiFi

pública, enquanto permite restrições mais relaxadas quando conectado à sua rede doméstica. Para um servidor, essas zonas não são tão importantes imediatamente porque o ambiente de rede raramente, ou nunca, muda.

Independentemente de quão dinâmico seu ambiente de rede possa ser, ainda é útil estar familiarizado com a ideia geral por trás de cada uma das zonas predefinidas para firewalld. Em ordem do menos confiável para o mais confiável , as zonas predefinidas são:

- **drop** - O nível mais baixo de confiança. Todas as conexões de entrada são interrompidas sem resposta e apenas as conexões de saída são possíveis.
- **block** - Semelhante ao anterior, mas em vez de simplesmente eliminar as conexões, as solicitações de entrada são rejeitadas com uma mensagem `icmp-host-prohibited` ou `icmp6-adm-prohibited`.
- **public** - Representa redes públicas não confiáveis. Você não confia em outros computadores, mas pode permitir conexões de entrada selecionadas caso a caso.
- **external** - Redes externas caso você esteja usando o firewall como gateway. Ele é configurado para mascaramento de NAT para que sua rede interna permaneça privada, mas acessível.
- **internal** - O outro lado da zona externa, usado para a parte interna de um gateway. Os computadores são bastante confiáveis e alguns serviços adicionais estão disponíveis.
- **dmz** - Usado para computadores localizados em uma DMZ (computadores isolados que não terão acesso ao resto da rede). Apenas certas conexões de entrada são permitidas.
- **work** - Usado para máquinas de trabalho. Confie na maioria dos computadores da rede. Mais alguns serviços podem ser permitidos.
- **home** - Um ambiente doméstico. Geralmente implica que você confia na maioria dos outros computadores e que mais alguns serviços serão aceitos.
- **trusted** - Confia em todas as máquinas da rede. A mais aberta das opções disponíveis e deve ser usada com moderação.

Para usar o firewall, podemos criar regras e alterar as propriedades de nossas zonas e, em seguida, atribuir nossas interfaces de rede às zonas mais apropriadas.

Permanência de regra

No firewalld, as regras podem ser designadas como permanentes ou imediatas. Se uma regra for adicionada ou modificada, por padrão, o comportamento do firewall em execução no momento será modificado. Na próxima inicialização, as modificações serão descartadas e as regras antigas serão aplicadas.

A maioria das operações `firewall-cmd` pode receber o sinalizador `--permanent` para indicar que o firewall não efêmero deve ser direcionado. Isso afetará o conjunto de regras que é recarregado na inicialização. Essa separação significa que você pode testar regras em sua instância de

firewall ativa e recarregar se houver problemas. Você também pode usar o sinalizador `--permanent` para construir um conjunto completo de regras ao longo do tempo, que serão aplicadas de uma vez quando o comando reload for emitido.

Instale e habilite seu firewall na inicialização

`firewalld` é instalado por padrão em algumas distribuições Linux, incluindo muitas imagens do CentOS 8. No entanto, pode ser necessário que você mesmo instale o firewalld:

```
[root@localhost joatham]# yum install firewalld
```

Depois de instalar `firewalld`, você pode habilitar o serviço e reinicializar o servidor. Lembre-se de que a ativação do firewalld fará com que o serviço comece na inicialização. É uma prática recomendada criar suas regras de firewall e aproveitar a oportunidade para testá-las antes de configurar esse comportamento para evitar possíveis problemas.

```
[root@localhost joatham]# systemctl enable firewalld
sudo reboot
```

Quando o servidor for reiniciado, seu firewall deve ser ativado, suas interfaces de rede devem ser colocadas nas zonas que você configurou (ou voltar para a zona padrão configurada), e quaisquer regras associadas às zonas serão aplicadas aos interfaces.

Podemos verificar se o serviço está em execução e acessível digitando:

```
[root@localhost joatham]# firewall-cmd --state
running
```

Isso indica que nosso firewall está instalado e funcionando com a configuração padrão.

Familiarizando-se com as regras atuais de firewall

Antes de começarmos a fazer modificações, devemos nos familiarizar com o ambiente padrão e as regras fornecidas pelo daemon.

Explorando os padrões

Podemos ver qual zona está atualmente selecionada como padrão digitando:

```
[root@localhost joatham]# firewall-cmd --get-default-zone
public
```

Como não demos nenhum comando para desviar da zona padrão e nenhuma de nossas interfaces está configurada para se ligar a outra zona, essa também será a única zona "ativa" (a zona que está controlando o tráfego para nossas interfaces). Podemos verificar isso digitando:

```
[root@localhost joatham]# firewall-cmd --get-active-zones
libvirt
  interfaces: virbr0
```

Como sabemos quais regras estão associadas à zona pública? Podemos imprimir a configuração da zona padrão digitando:

```
[root@localhost joatham]# firewall-cmd --list-all
You're performing an operation over default zone ('public'),
but your connections/interfaces are in zone 'libvirt' (see --get-active-zones)
You most likely need to use --zone=libvirt option.

public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Explorando Zonas Alternativas

Agora temos uma boa ideia sobre a configuração da zona padrão e ativa. Podemos encontrar informações sobre outras zonas também.

Para obter uma lista das zonas disponíveis, digite:

```
[root@localhost joatham]# firewall-cmd --get-zones
block dmz drop external home internal libvirt nm-shared public trusted work
```

Podemos ver a configuração específica associada a uma zona, incluindo o parâmetro `--zone=` em nosso comando `--list-all`:

```
[root@localhost joatham]# firewall-cmd --zone=home --list-all
home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client mdns samba-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Você pode gerar todas as definições de zona usando a opção `--list-all-zones`. Provavelmente, você desejará canalizar a saída para um pager para facilitar a visualização:

```
[root@localhost joatham]# firewall-cmd --list-all-zones | less
```

Seleção de zonas para suas interfaces

A menos que você tenha configurado suas interfaces de rede de outra forma, cada interface será colocada na zona padrão quando o firewall for inicializado.

Mudando a zona de uma interface

Você pode fazer a transição de uma interface entre zonas durante uma sessão usando o parâmetro `--zone=` em combinação com o parâmetro `--change-interface=`. Como acontece com todos os comandos que modificam o firewall, você precisará usar sudo.

Por exemplo, podemos fazer a transição de nossa interface `enp0s3` para a zona `home` digitando o seguinte:

```
[root@localhost joatham]# firewall-cmd --zone=home --change-interface=enp0s3
```

```
success
```

Podemos verificar se isso foi bem-sucedido solicitando as zonas ativas novamente:

```
[root@localhost joatham]# firewall-cmd --get-active-zones
home
  interfaces: enp0s3
libvirt
  interfaces: virbr0
```

Ajustando a zona padrão

Se todas as suas interfaces podem ser gerenciadas melhor por uma única zona, provavelmente é mais fácil selecionar a melhor zona padrão e, em seguida, usá-la para sua configuração.

Você pode alterar a zona padrão com o parâmetro `--set-default-zone=`. Isso mudará imediatamente qualquer interface que tenha voltado ao padrão para a nova zona:

```
[root@localhost joatham]# firewall-cmd --set-default-zone=home
```

Definindo regras para seus aplicativos

A maneira básica de definir exceções de firewall para os serviços que você deseja disponibilizar é bastante direta. Examinaremos a ideia básica aqui.

Adicionando um serviço às suas zonas

O método mais simples é adicionar os serviços ou portas de que você precisa às zonas que está usando. Novamente, você pode obter uma lista dos serviços disponíveis com a opção `--get-services`:

```
[root@localhost joatham]# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client amqp amqps apcupsd audit bacula bacula-client
      bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph
     ceph-mon cfengine cockpit collectd condor-collector ctdb dhcp dhcpv6 dhcpv6-client
     distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch
     etcd-client etcd-server finger freeipa-4 freeipa-ldap freeipa-ldaps freeipa-
     replication freeipa-trust ftp galera ganglia-client ganglia-master git grafana gre
     high-availability http https imap imaps ipp ipp-client ipsec irc ircs iscsi-target
     isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-
     apiserver ldap ldaps libvirt libvirt-tls lightning-network llmnr managesieve matrix
     mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nfs
```

```
 nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-
vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy
prometheus proxy-dhcp ptp pulseaudio puppetmaster quassel radius rdp redis redis-
sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane
 sip sips slp smtp smtp-submission smtps snmp snmptrap spideroak-lansync spotify-sync
squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui synergy syslog syslog
-tls telnet tentacle tftp tftp-client tile38 tinc tor-socks transmission-client upnp-
client vdsm vnc-server wbem-http wbem-https wsman wsmans xdmcp xmpp-bosh xmpp-client
xmpp-local xmpp-server zabbix-agent zabbix-server
```

Você pode habilitar um serviço para uma zona usando o parâmetro `--add-service=`. A operação terá como alvo a zona padrão ou qualquer zona especificada pelo parâmetro `--zone=`. Por padrão, isso só ajustará a sessão de firewall atual. Você pode ajustar a configuração do firewall permanente incluindo a flag `--permanent`.

Por exemplo, se estivermos executando um servidor web servindo tráfego HTTP convencional, podemos permitir esse tráfego para interfaces em nossa zona `public` para esta sessão, digitando:

```
[root@localhost joatham]# firewall-cmd --zone=public --add-service=http
success
```

Você pode omitir `--zone=` se deseja modificar a zona padrão. Podemos verificar se a operação foi bem-sucedida usando as operações `--list-all` ou `--list-services`:

```
[root@localhost joatham]# firewall-cmd --zone=public --list-services
cockpit dhcpv6-client http ssh
```

Depois de testar se tudo está funcionando como deveria, você provavelmente desejará modificar as regras de firewall permanentes para que seu serviço ainda esteja disponível após uma reinicialização. Podemos tornar nossa mudança de zona `public` permanente digitando:

```
[root@localhost joatham]# firewall-cmd --zone=public --permanent --add-service=http
success
```

Você pode verificar se isso foi bem-sucedido adicionando o sinalizador `--permanent` à operação `--list-services`. Você precisa usar sudo para qualquer operação `--permanent`:

```
[root@localhost joatham]# firewall-cmd --zone=public --permanent --list-services
cockpit dhcpv6-client http ssh
```

Sua zona `public` agora permitirá o tráfego HTTP da web na porta `80`. Se o seu servidor da web estiver configurado para usar `SSL/TLS`, você também desejará adicionar o serviço `https`. Podemos adicionar isso à sessão atual e ao conjunto de regras permanente digitando:

```
[root@localhost joatham]# firewall-cmd --zone=public --add-service=https
success
[root@localhost joatham]# firewall-cmd --zone=public --permanent --add-service=https
success
```

Criando Suas Próprias Zonas

Embora as zonas predefinidas provavelmente sejam mais do que suficientes para a maioria dos usuários, pode ser útil definir suas próprias zonas que são mais descritivas de suas funções.

Por exemplo, você pode querer criar uma zona para o seu servidor web, chamada "publicweb". No entanto, você pode querer ter outra zona configurada para o serviço DNS que você fornece em sua rede privada. Você pode querer uma zona chamada "privateDNS" para isso.

Ao adicionar uma zona, você deve adicioná-la à configuração de firewall permanente. Você pode então recarregar para trazer a configuração para sua sessão de execução. Por exemplo, poderíamos criar as duas zonas que discutimos acima digitando:

```
[root@localhost joatham]# firewall-cmd --permanent --new-zone=web_publica
success
[root@localhost joatham]# firewall-cmd --permanent --new-zone=dns_privada
success
```

Você pode verificar se eles estão presentes em sua configuração permanente digitando:

```
[root@localhost joatham]# firewall-cmd --permanent --get-zones
block dmz dns_privada drop external home internal libvirt nm-shared public trusted
     web_publica work
```

Conforme declarado antes, eles ainda não estarão disponíveis na instância atual do firewall:

```
[root@localhost joatham]# firewall-cmd --get-zones
block dmz drop external home internal libvirt nm-shared public trusted work
```

Recarregue o firewall para trazer essas novas zonas para a configuração ativa:

```
[root@localhost joatham]# firewall-cmd --reload
success
[root@localhost joatham]# firewall-cmd --get-zones
block dmz dns_privada drop external home internal libvirt nm-shared public trusted
      web_publica work
```

Agora, você pode começar a atribuir os serviços e portas apropriados às suas zonas. Normalmente, é uma boa ideia ajustar a instância ativa e, em seguida, transferir essas alterações para a configuração permanente após o teste. Por exemplo, para a zona `web_publica`, você pode querer adicionar os serviços `SSH`, `HTTP` e `HTTPS`:

```
[root@localhost joatham]# firewall-cmd --zone=web_publica --add-service=ssh
success
[root@localhost joatham]# firewall-cmd --zone=web_publica --add-service=http
success
[root@localhost joatham]# firewall-cmd --zone=web_publica --add-service=https
success
[root@localhost joatham]# firewall-cmd --zone=web_publica --list-all
web_publica
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: http https ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Da mesma forma, podemos adicionar o serviço DNS à nossa zona “privateDNS”:

```
[root@localhost joatham]# firewall-cmd --zone=dns_privada --add-service=dns
success
[root@localhost joatham]# firewall-cmd --zone=dns_privada --list-all
dns_privada
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dns
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
```

```
  icmp-blocks:
  rich rules:
```

Poderíamos então mudar nossas interfaces para essas novas zonas para testá-las:

```
[root@localhost joatham]# firewall-cmd --zone=web_publica --change-interface=enp0s3
success
[root@localhost joatham]# firewall-cmd --zone=dns_privada --change-interface=enp0s8
```

Neste ponto, você tem a oportunidade de testar sua configuração. Se esses valores funcionarem para você, adicione as mesmas regras à configuração permanente. Você pode fazer isso reaplicando as regras com o sinalizador `--permanent`:

```
[root@localhost joatham]# firewall-cmd --zone=web_publica --permanent --add-service=ssh
[root@localhost joatham]# firewall-cmd --zone=web_publica --permanent --add-service=http
[root@localhost joatham]# firewall-cmd --zone=web_publica --permanent --add-service=https
[root@localhost joatham]# firewall-cmd --zone=privateDNS --permanent --add-service=dns
```

Depois de aplicar permanentemente essas regras, você pode reiniciar sua rede e recarregar o serviço de firewall:

```
sudo systemctl restart network
sudo systemctl reload firewalld
```

Valide se as zonas corretas foram atribuídas:

```
firewall-cmd --get-active-zones
```

E valide se os serviços apropriados estão disponíveis para ambas as zonas:

```
[root@localhost joatham]# firewall-cmd --zone=web_publica --list-services
http https ssh
```

```
[root@localhost joatham]# firewall-cmd --zone=dns_privada --list-services
dns
```

Você configurou com sucesso suas próprias zonas! Se você deseja tornar uma dessas zonas o padrão para outras interfaces, lembre-se de configurar esse comportamento com o parâmetro `--set-default-zone=`:

```
[root@localhost joatham]# firewall-cmd --set-default-zone=publicweb
```

Conclusão

Agora você deve ter um bom entendimento de como administrar o serviço `firewalld` em seu sistema `RHEL` para uso diário.

O serviço `firewalld` permite configurar regras de manutenção e conjuntos de regras que levam em consideração seu ambiente de rede. Ele permite que você faça a transição perfeita entre diferentes políticas de firewall por meio do uso de zonas e oferece aos administradores a capacidade de abstrair o gerenciamento de portas em definições de serviço mais amigáveis. Adquirir um conhecimento prático deste sistema permitirá que você aproveite a flexibilidade e o poder que esta ferramenta oferece. Além disso, o mais importante: que consiga passar na prova `RHCSA`.# Criar e usar listas de controle de acesso a arquivos ## Pontos de estudo para o exame Os candidatos ao exame `RHCSA` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar segurança**
  - Defina as configurações do firewall usando firewall-cmd/firewalld
  - **Crie e use listas de controle de acesso a arquivos**
  - Configure a autenticação baseada em chave para SSH
  - Defina modos de aplicação e permissivos para SELinux
  - Liste e identifique o contexto de processos e arquivos do SELinux
  - Restaure contextos de arquivo padrão
  - Use as configurações booleanas para modificar as configurações do SELinux do sistema
  - Diagnostique e resolva violações de rotina da política do SELinux

Introdução

Pode levar algum tempo para se acostumar, mas as listas de controle de acesso do Linux, ou `ACLs`, são inestimáveis para obter um controle mais refinado das permissões do sistema de arquivos do Linux.

Revendo o básico

O sistema de arquivos Linux nos dá **três** tipos de permissões. Aqui está uma revisão simplificada:

- **U** ser (ou proprietário do usuário)
- **G** rupo (ou grupo proprietário)
- **O** outro (todos os outros)

Com essas permissões, podemos conceder três tipos de acesso:

- **R** ead
- **W** rite
- e **X** ecute

Esses níveis de acesso costumam ser adequados em muitos casos. O sistema padrão de permissão de acesso para arquivos e diretórios no Linux pode ser extendido com o uso da ACL – *Access Control List*.

As ACLs reduzem um pouco a performance do seu sistema. Na partição do sistema (/), as permissões padrões são suficientes. Ative a ACL em partições específicas como a home e crie uma para a sua necessidade.

Através das **Listas de Controle de Acesso** podemos estender essa capacidade adicionando mais donos, grupos e outros mesmo que não correspondam aos originais. Alguns dos sistemas de arquivos que possuem suporte para ACLs são: - **Ext2** - **Ext3** - **Ext4** - **ReiserFS** - **JFS** - **XFS**

A ACL é um recurso do kernel Linux. Para verificar se o sistema de arquivos possui essa característica, use o comando abaixo:

```
[root@rhel8 joatham]# tune2fs -l /dev/sdb | grep "Default mount options:"
Default mount options:    user_xattr acl
```

> Caso não seja exibida a opção acl, edite /etc/fstab e acrescente acl em options. Talvez seja necessário instalar o pacote e reiniciar o sistema. Porém, no RHEL, isso já vem funcionando por padrão.

Utilizando ACL

Existem apenas dois tipos de ACLs:

- **Regras de Acesso** – Especificam as informações de acesso para um arquivo ou diretório único
- **Regras Padrão** – Aplicadas apenas a diretórios e especificam informações de acesso padronizadas para todos os arquivos dentro do diretório.

Arquivos que não possuem regras herdam a configuração do diretório pai com base nas permissões padrão.

Conforme brevemente explicado anteriormente, é possível definir valores padrões para usuários, grupos e outros, em pastas e arquivos, diferente do dono, grupo e outros padrão do sistema de arquivos.

- **getfacl** – Utilizado para visualizar as permissões;
- **setfacl** – Utilizado para alterar as permissões.

Hands On

> Os comandos serão executados no diretório corrente `/mnt`.

```
[root@rhel8 joatham]# mount /dev/sdb /mnt/acl/
```

Vamos acessar a partição e criar um arquivo para realizarmos os testes de `acl`:

```
[root@rhel8 acl]# cd /mnt/acl/4LINUX
[root@rhel8 4LINUX]# vi relatorio.txt
```

- Verifique as permissões de acesso ao arquivo `ls -l`:

```
[root@rhel8 4LINUX]# ls -l
total 4
-rw-r--r--. 1 root root 19 out 15 14:33 relatorio.txt
```

- Verifique as ACLs do arquivo usando o comando `getfacl`:

```
[root@rhel8 4LINUX]# getfacl relatorio.txt
# file: relatorio.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

As primeiras três linhas do comando `getfacl` constituem o cabeçalho: - Nome do arquivo; - Proprietário; - Grupo ao qual ele está associado.

Na linha que inicia com `user`, os símbolos `::` indicam que a linha especifica as permissões do proprietário do arquivo. De forma análoga, a linha `group::` indica as permissões do grupo associado ao arquivo, e a linha `other::`, as permissões dos outros. Nenhum nome pode ser associado a `other`.

- Adicionar ou modificar regras de permissões ao arquivo usando o comando `setfacl` com a flag `--modify` (ou `-m`), com a sintaxe a seguir:

```
# setfacl -m ugo:nome:permissões lista-de-arquivos
```

Onde:

- **u** - Ajusta permissão de um usuário qualquer.
- **g** - Ajusta permissão de um grupo qualquer.
- **o** - Ajusta permissão de todos os outros usuários.
- **nome** - Nome do usuário ou grupo para o qual iremos ajustar as permissões.
- **permissões** - Permissões no modo simbólico ou numérico (absoluto)
- **lista-de-arquivos** - Lista de arquivos aos quais as permissões serão aplicadas.

> O comando `setfacl` funciona como uma espécie de `chmod` para `acls`.

Ao especificarmos permissões para os outros usuários (others), devemos omitir o parâmetro nome do comando.

As permissões simbólicas usam as letras `r`, `w`, `x` para representar as permissões de arquivo, ao passo que as permissões absolutas usam um número octal (de `0` a `7`).

O arquivo relatorio.txt será configurado para ter as seguintes permissões:

- **Usuário proprietário**: `rw`
- **Grupo do arquivo**: `rw`
- **Outros**: sem permissão nenhuma
- **Usuário joatham**: somente leitura

Veja que com o esquema de permissões padrão do Linux não conseguiríamos aplicar esse conjunto de permissões da forma como necessitamos. Se o usuário joatham for membro do grupo do arquivo, terá permissão rw, e se for "outro" usuário, não terá permissão alguma. Vamos usar então a ACL para dar essa permissão a ele. Primeiramente, damos as permissões comuns usando o comando `chmod`:

```
[root@rhel8 4LINUX]# chmod 600 relatorio.txt
[root@rhel8 4LINUX]# ls -l
total 4
-rw-------. 1 root root 19 out 15 14:33 relatorio.txt
```

- Ajuste a permissão do usuário joatham usando o comando `setfacl`:

```
[root@rhel8 4LINUX]# setfacl -m u:joatham:r relatorio.txt
```

- E verificamos o resultado com `getfacl`:

```
[root@rhel8 4LINUX]# getfacl relatorio.txt
# file: relatorio.txt
# owner: root
# group: root
user::rw-
user:joatham:r--
group::---
mask::r--
other::---
```

> Observe que o usuário joatham aparece na saída do comando `getfacl`, com a permissão `r`(leitura) definida.

- Vamos conferir a saída do comando `ls -l` agora:

```
[root@rhel8 4LINUX]# ls -l
total 8
-rw-r-----+ 1 root root 19 out 15 14:33 relatorio.txt
```

> Note que há agora um sinal de adição (+) ao lado das permissões do arquivo. Esse é um flag que indica permissões definidas por ACL neste arquivo.

Podemos configurar regras de ACL para mais de um usuário ao mesmo tempo usando o comando `setfacl`. Por exemplo, vamos dar as permissões `rwx` para o usuário joatham e somente leitura (`r`) para o usuário pedro. Para isso, separamos as configurações de cada usuário com uma vírgula:

```
[root@rhel8 4LINUX]# setfacl -m u:joatham:rwx,u:pedro:r-- relatorio.txt
```

- Analise o resultado com o comando `getfacl`:

```
[root@rhel8 4LINUX]# getfacl relatorio.txt
# file: relatorio.txt
# owner: root
# group: root
user::rw-
user:joatham:rwx
user:pedro:r--
group::---
mask::rwx
other::---
```

Também podemos aplicar regras de `acl` a um grupo específico usando a opção `g`, como por exemplo um grupo de nome `diretores`:

```
[root@rhel8 4LINUX]# setfacl -m g:diretores:rwx relatorio.txt
```

Ou ainda aplicar as regras de `acl` a mais de um arquivo simultaneamente, separando seus nomes por espaços em branco:

```
[root@rhel8 4LINUX]# setfacl -m g:diretores:rwx relatorio.txt planilhas.txt
```

Para remover as regras de ACL de um usuário ou grupo, use a opção `-x`. Esta opção não age

sobre o proprietário do arquivo ou sobre o grupo associado ao arquivo.

- Remover a permissão de leitura que foi atribuída ao usuário pedro no arquivo relatorio.txt:

```
[root@rhel8 4LINUX]# setfacl -x u:pedro relatorio.txt
```

- Verifique o resultado. Use a opção `--omit-header` para que o comando `getfacl` não mostre o cabeçalho na saída:

```
[root@rhel8 4LINUX]# getfacl --omit-header relatorio.txt
user::rw-
user:joatham:rwx
group::---
group:diretores:rwx
mask::rwx
other::---
```

Veja que a permissão atribuída anteriormente ao usuário pedro desapareceu. Não devemos especificar as permissões a serem retiradas quando usamos a opção `-x` – especifique apenas o valor `ugo` e o nome do item.

Podemos também retirar todas as regras de `ACL` aplicadas a um arquivo de uma vez. Para isso, use a opção `-b`, seguida do nome do arquivo:

```
[root@rhel8 4LINUX]# setfacl -b relatorio.txt
[root@rhel8 4LINUX]# getfacl relatorio.txt
# file: relatorio.txt
# owner: root
# group: root
user::rw-
group::---
other::---
```

> Veja que a permissão atribuída ao usuário joatham foi excluída.

Configurando regras padrão em um diretório

Podemos configurar regras de `ACL` padrão em um diretório, de modo que essas regras sejam aplicadas automaticamente a todos os arquivos dentro desse diretório que não possuam suas

próprias ACLs explícitas.

Já temos o diretório criado chamado 4LINUX, mas vamos criar também dois grupos: - grupo diretores - grupo gerentes

```
[root@rhel8 4LINUX]# groupadd diretores
[root@rhel8 4LINUX]# groupadd gerentes
```

Agora, vamos criar as seguintes regras padrão a serem aplicadas no diretório 4LINUX: os membros do grupo **gerentes** terão permissões de leitura e execução (r-x) e os membros do grupo **diretores** terão permissões de leitura, escrita e execução (rwx). A opção -d nos permite criar as regras padrão no diretório:

```
[root@rhel8 acl]# setfacl -d -m g:gerentes:r-x,g:diretores:rwx 4LINUX/
```

- Confira se as ACLs foram aplicadas corretamente:

```
[root@rhel8 acl]# getfacl 4LINUX/
# file: 4LINUX/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:diretores:rwx
default:group:gerentes:r-x
default:mask::rwx
default:other::r-x
```

Cada uma das regras padrão criadas começa com a palavra default: na listagem retornada pelo comando getfacl. Essas regras serão aplicadas aos arquivos dentro deste diretório que não possuírem regras de ACL explícitas. Podemos também criar regras de ACL para o diretório em si.

Vamos entrar no diretório 4LINUX, criar um arquivo de nome importante.txt e verificar as ACLs que ele herdará do diretório:

```
[root@rhel8 acl]# cd 4LINUX/
[root@rhel8 4LINUX]# touch importante.txt
[root@rhel8 4LINUX]# getfacl importante.txt
```

```
# file: importante.txt
# owner: root
# group: root
user::rw-
group::r-x          #effective:r--
group:diretores:rwx     #effective:rw-
group:gerentes:r-x      #effective:r--
mask::rw-
other::r--
```

Note que o arquivo recebeu as permissões padrão aplicadas no diretório onde ele se encontra, e note também as permissões efetivas que ele possui, listadas na coluna à direita na listagem.

As permissões efetivas são o resultado da aplicação da máscara de direitos efetivos sobre as permissões aplicadas nos usuários e grupos configurados na `ACL`.

Remover as permissões padrão de um diretório

É possível também remover apenas as permissões padrão de um diretório, sem no entanto excluir permissões eventualmente aplicadas a usuários ou grupos específicos no diretório. Para isso empregamos a opção `k`.

Vamos testar isso aplicando permissões padrão ao diretório `4LINUX` e, logo após, aplicando permissão específica ao usuário `pedro`, leitura e escrita, no diretório:

```
[root@rhel8 acl]# groupadd ti
[root@rhel8 acl]# setfacl -d -m g:ti:r-x,g:diretores:rwx 4LINUX/
[root@rhel8 acl]# setfacl -m u:pedro:rw 4LINUX/
```

- Verificando as permissões aplicadas:

```
[root@rhel8 acl]# getfacl 4LINUX/
# file: 4LINUX/
# owner: root
# group: root
user::rwx
user:pedro:rw-
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:group::r-x
default:group:diretores:rwx
default:group:gerentes:r-x
default:group:ti:r-x
default:mask::rwx
default:other::r-x
```

- Vamos agora retirar as opções padrão do diretório:

[root@rhel8 acl]# setfacl -k 4LINUX/

- Verificando:

```
[root@rhel8 acl]# getfacl 4LINUX/
# file: 4LINUX/
# owner: root
# group: root
user::rwx
user:pedro:rw-
group::r-x
mask::rwx
other::r-x
```

Perceba que as permissões padrão do diretório sumiram, mas a permissão aplicada ao usuário pedro permaneceu.

Máscara de Direitos Efetiva (Mask)

A linha que aparece na saída do comando `getfacl` e que começa com a palavra `mask` especifica a máscara de direitos efetiva. Esta máscara limita as permissões efetivas garantidas aos grupos e usuários da `ACL`.

Ela não afeta o usuário proprietário do arquivo, nem seu grupo associado. Em outras palavras, a máscara não afeta as permissões tradicionais do Linux.

Porém, o comando `setfacl` sempre ajusta a máscara de direitos efetiva para o nível de permissões de `ACL` menos restritivo para o arquivo. A máscara não terá efeito a não ser que seja configurada explicitamente após criarmos as regras de `ACL` no arquivo.

Para isso, usamos a palavra `mask` no lugar de `ugo` e não especificamos um nome no comando `setfacl`.

Por exemplo, vamos configurar a máscara de direitos efetiva para somente leitura no arquivo `planilhas.txt`:

```
[root@rhel8 acl]# setfacl -m mask::r-- 4LINUX/planilhas.txt
```

- Vamos ver agora como ficaram as permissões no arquivo `planilhas.txt`:

```
[root@rhel8 acl]# getfacl 4LINUX/planilhas.txt
# file: 4LINUX/planilhas.txt
# owner: root
# group: root
user::rw-
group::r--
group:diretores:rwx     #effective:r--
mask::r--
other::r--
```

Veja que, embora os grupos `diretores` e `gerentes` tenham permissões `r-x` e `rwx`, respectivamente, suas permissões efetivas são calculadas com base na máscara, mostrada na linha `mask::r--`, e no final a permissão dos grupos é `r--` (somente leitura).

As permissões do proprietário e do grupo associado do arquivo não foram alteradas.

28

Configure a autenticação baseada em chave para SSH

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar segurança**
  - Defina as configurações do firewall usando firewall-cmd/firewalld
  - Crie e use listas de controle de acesso a arquivos
  - **Configure a autenticação baseada em chave para SSH**
  - Defina modos de aplicação e permissivos para SELinux
  - Liste e identifique o contexto de processos e arquivos do SELinux
  - Restaure contextos de arquivo padrão
  - Use as configurações booleanas para modificar as configurações do SELinux do sistema
  - Diagnostique e resolva violações de rotina da política do SELinux

Introdução

O SSH, ou shell seguro, é um protocolo criptografado usado para administrar e se comunicar com servidores. Ao trabalhar com um servidor Linux, existem boas chances de você gastar a

maior parte do seu tempo em uma sessão de terminal conectada ao seu servidor através do SSH.

Embora existam outras maneiras de fazer login em um servidor SSH, neste guia, vamos focar na configuração de chaves SSH. As chaves SSH oferecem uma maneira fácil e extremamente segura de fazer login no seu servidor. Por esse motivo, este é o método que recomendamos para todos os usuários.

Como as chaves SSH funcionam?

Um servidor SSH pode autenticar clientes usando uma variedade de métodos diferentes. O mais básico deles é a autenticação por senha, que embora fácil de usar, não é o mais seguro.

Apesar de as senhas serem enviadas ao servidor de maneira segura, elas geralmente não são complexas ou longas o suficiente para resistirem a invasores persistentes. O poder de processamento moderno combinado com scripts automatizados torna possível forçar a entrada de maneira bruta em uma conta protegida por senha. Mesmo que existam outros métodos para adicionar segurança adicional (`fail2ban`, etc), as chaves SSH são comprovadamente uma alternativa confiável e segura.

Os pares de chaves SSH são duas chaves criptografadas e seguras que podem ser usadas para autenticar um cliente em um servidor SSH. Cada par de chaves consiste em uma chave pública e uma chave privada.

A chave privada é mantida pelo cliente e deve ser mantida em absoluto sigilo. Qualquer comprometimento da chave privada permitirá que o invasor faça login em servidores que estejam configurados com a chave pública associada sem autenticação adicional. Como uma forma de precaução adicional, a chave pode ser criptografada em disco com uma frase secreta.

A chave pública associada pode ser compartilhada livremente sem consequências negativas. A chave pública pode ser usada para criptografar mensagens que apenas a chave privada pode descriptografar. Essa propriedade é usada como uma maneira de autenticar usando o par de chaves.

A chave pública é enviada a um servidor remoto de sua preferência para que você possa fazer login via SSH. Ela é adicionada a um arquivo especial dentro da conta de usuário em que você estará fazendo login, chamado `~/.ssh/authorized_keys`.

Quando um cliente tenta autenticar usando chaves SSH, o servidor testa o cliente para verificar se ele tem posse da chave privada. Se o cliente puder provar que possui a chave privada, a sessão do shell é gerada ou o comando solicitado é executado.

Como criar chaves SSH

O primeiro passo para configurar a autenticação de chaves SSH para seu servidor é gerar um par de chaves SSH no seu computador local.

Para fazer isso, podemos usar um utilitário especial chamado `ssh-keygen`, que vem incluso com o conjunto padrão de ferramentas do OpenSSH. Por padrão, isso criará um par de chaves RSA de 2048 bits, que é suficiente para a maioria dos usos.

No seu computador local, gere um par de chaves SSH digitando:

```
[root@localhost joatham]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

O utilitário solicitará que seja selecionado um local para as chaves que serão geradas. Por padrão, as chaves serão armazenadas no diretório `~/.ssh` dentro do diretório home do seu usuário. A chave privada será chamada de `id_rsa` e a chave pública associada será chamada de `id_rsa.pub`.

Normalmente, é melhor manter e utilizar o local padrão neste estágio. Fazer isso permitirá que seu cliente SSH encontre automaticamente suas chaves SSH ao tentar autenticar-se. Se quiser escolher um caminho não padrão, digite-o agora. Caso contrário, pressione ENTER para aceitar o padrão.

Caso tenha gerado um par de chaves SSH anteriormente, pode ser que você veja um prompt parecido com este:

```
/home/`username`/.ssh/id_rsa already exists.
Overwrite (y/n)?
```

Se escolher substituir a chave no disco, você **não** poderá autenticar-se usando a chave anterior. Seja cuidadoso ao selecionar o **sim**, uma vez que este é um processo destrutivo que não pode ser revertido.

```
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

Em seguida, você será solicitado a digitar uma frase secreta para a chave. Esta é uma frase secreta opcional que pode ser usada para criptografar o arquivo de chave privada no disco.

Você pode estar se perguntando sobre quais são as vantagens que uma chave SSH oferece se ainda é necessário digitar uma frase secreta. Algumas das vantagens são:

- A chave SSH privada (a parte que pode ser protegida por uma frase secreta), nunca é exposta na rede. A frase secreta é usada apenas para descriptografar a chave na máquina local. Isso significa que utilizar força bruta na rede não será possível contra a frase secreta.
- A chave privada é mantida dentro de um diretório restrito. O cliente SSH não reconhecerá chaves privadas que não são mantidas em diretórios restritos. A chave em si também precisa ter permissões restritas (leitura e gravação apenas disponíveis para o proprietário). Isso significa que outros usuários no sistema não podem bisbilhotar.
- Qualquer invasor que queira decifrar a frase secreta da chave SSH privada precisa já ter acesso ao sistema. Isso significa que eles já terão acesso à sua conta de usuário ou conta root. Se você estiver nesta posição, a frase secreta pode impedir que o invasor faça login imediatamente em seus outros servidores. Espera-se que isso dê a você tempo suficiente para criar e implementar um novo par de chaves SSH e remover o acesso da chave comprometida.

Como a chave privada nunca é exposta à rede e é protegida através de permissões de arquivos, este arquivo nunca deve ser acessível a qualquer um que não seja você (e o usuário root). A frase secreta serve como uma camada adicional de proteção caso essas condições sejam comprometidas.

Uma frase secreta é uma adição opcional. Se você inserir uma, será necessário fornecê-la sempre que for usar essa chave (a menos que você esteja executando um software de agente SSH que armazena a chave descriptografada). Recomendamos a utilização de uma frase secreta, mas se você não quiser definir uma, basta pressionar ENTER para ignorar este prompt.

```
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:togsJ72aQX6WDRQPcotKwUYYVI2N3NS+m35Hh1qWxLM root@localhost.localdomain
The key's randomart image is:
+---[RSA 3072]----+
|*=+.@..          |
|.o.B O .         |
|... o o   .      |
|.. .   .   +     |
|. . .   S . =    |
| o o = + . E .   |
|  = O o + = .    |
|   O . o o .     |
|  o.. ... .      |
```

```
+----[SHA256]-----+
```

Agora, você tem uma chave pública e privada que pode usar para se autenticar. O próximo passo é colocar a chave pública no seu servidor para que você possa usar a autenticação baseada em chaves SSH para fazer login.

Como copiar uma chave pública para seu servidor

Se você já tiver um servidor disponível e não incorporou chaves em sua criação, ainda é possível enviar sua chave pública e usá-la para autenticar-se no seu servidor.

O método a ser usado depende em grande parte das ferramentas disponíveis e dos detalhes da sua configuração atual. Todos os métodos a seguir geram o mesmo resultado final. O método mais fácil e automatizado é o primeiro e cada método depois dele necessita de passos manuais adicionais se você não conseguir usar os métodos anteriores.

Copiando sua chave pública usando o SSH-Copy-ID

A maneira mais fácil de copiar sua chave pública para um servidor existente é usando um utilitário chamado `ssh-copy-id`. Por conta da sua simplicidade, este método é recomendado se estiver disponível.

A ferramenta `ssh-copy-id` vem inclusa nos pacotes OpenSSH em muitas distribuições, de forma que você pode tê-la disponível em seu sistema local. Para que este método funcione, você já deve ter acesso via SSH baseado em senha ao seu servidor.

Para usar o utilitário, você precisa especificar apenas o host remoto ao qual gostaria de se conectar e a conta do usuário que tem acesso SSH via senha. Esta é a conta na qual sua chave SSH pública será copiada.

A sintaxe é:

```
[root@localhost joatham]# ssh-copy-id kali@10.11.10.152
```

Pode ser que apareça uma mensagem como esta:

```
The authenticity of host '10.11.10.152 (10.11.10.152)' can't be established.
ECDSA key fingerprint is SHA256:NjFhUCBpj+9qgnh1qlhHNmJPJOqqip1pVwfqsFWtjdA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Isso significa que seu computador local não reconhece o host remoto. Isso acontecerá na primeira vez que você se conectar a um novo host. Digite `yes` e pressione `ENTER` para continuar.

Em seguida, o utilitário irá analisar sua conta local em busca da chave `id_rsa.pub` que criamos mais cedo. Quando ele encontrar a chave, irá solicitar a senha da conta do usuário remoto:

```
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
     that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
     is to install the new keys
kali@10.11.10.152's password:
```

Digite a senha (sua digitação não será exibida para fins de segurança) e pressione ENTER. O utilitário se conectará à conta no host remoto usando a senha que você forneceu. Então, ele copiará o conteúdo da sua chave `~/.ssh/id_rsa.pub` em um arquivo no diretório da conta remota home `~/.ssh` chamado `authorized_keys`.

Você verá um resultado que se parece com este:

```
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'kali@10.11.10.152'"
and check to make sure that only the key(s) you wanted were added.
```

Neste ponto, sua chave `id_rsa.pub` foi enviada para a conta remota. Continue para a próxima seção.

Copiando sua chave pública usando o SSH

Se não tiver o `ssh-copy-id` disponível, mas tiver acesso SSH baseado em senha a uma conta do seu servidor, você pode fazer o upload das suas chaves usando um método `SSH` convencional.

É possível fazer isso resgatando o conteúdo da nossa chave `SSH` pública do nosso computador local e enviando-o através de uma conexão via protocolo `SSH` ao servidor remoto. Do outro lado, certificamo-nos de que o diretório `~/.ssh` existe na conta que estamos usando e então enviamos o conteúdo recebido em um arquivo chamado `authorized_keys` dentro deste diretório.

Vamos usar o símbolo de redirecionamento `>>` para adicionar o conteúdo ao invés de substituí-lo. Isso permitirá que adicionemos chaves sem destruir chaves previamente adicionadas.

O comando completo ficará parecido com este:

```
cat ~/.ssh/id_rsa.pub | ssh username@remote_host "mkdir -p ~/.ssh && cat >> ~/.ssh/
    authorized_keys"
```

Pode ser que apareça uma mensagem como esta:

```
The authenticity of host '111.111.11.111 (111.111.11.111)' can't be established.
ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:fe.
Are you sure you want to continue connecting (yes/no)? yes
```

Isso significa que seu computador local não reconhece o host remoto. Isso acontecerá na primeira vez que você se conectar a um novo host. Digite `yes` e pressione `ENTER` para continuar.

Depois disso, você será solicitado a inserir a senha da conta na qual está tentando se conectar:

```
username@111.111.11.111's password:
```

Após digitar sua senha, o conteúdo da sua chave `id_rsa.pub` será copiado para o final do arquivo `authorized_keys` da conta do usuário remoto. Continue para a próxima seção se o processo foi bem-sucedido.

Autenticar-se em seu servidor usando chaves SSH

Se tiver completado um dos procedimentos acima, você deve conseguir fazer login no host remoto sem a senha da conta remota.

O processo básico é o mesmo:

```
[root@localhost joatham]# ssh kali@10.11.10.152
```

Se essa é a primeira vez que você se conecta a este host (caso tenha usado o último método acima), pode ser que veja algo como isso:

```
The authenticity of host '111.111.11.111 (111.111.11.111)' can't be established.
ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:fe.
Are you sure you want to continue connecting (yes/no)? yes
```

Isso significa que seu computador local não reconhece o host remoto. Digite `yes` e então pressione ENTER para continuar.

Se não forneceu uma frase secreta para sua chave privada, você será logado imediatamente.

```
[root@localhost joatham]# ssh kali@10.11.10.152
Linux kali 5.7.0-kali1-amd64 #1 SMP Debian 5.7.6-1kali2 (2020-07-01) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
kali@kali:~$
```

Se forneceu uma frase secreta para a chave privada quando a criou, você será solicitado a digitá-la agora. Depois disso, uma nova sessão de shell deve ser-lhe gerada com a conta no sistema remoto.

Caso isso dê certo, continue para descobrir como bloquear o servidor.

Desativando a autenticação por senha no seu servidor

Se conseguiu logar na sua conta usando o SSH sem uma senha, você configurou com sucesso a autenticação baseada em chaves SSH na sua conta. Entretanto, seu mecanismo de autenticação baseado em senha ainda está ativo, o que significa que seu servidor ainda está exposto a ataques por força bruta.

Antes de completar os passos, certifique-se de que você tenha uma autenticação baseada em chaves SSH configurada para a conta root neste servidor, ou, de preferência, que tenha uma autenticação baseada em chaves SSH configurada para uma conta neste servidor com privilégios `sudo`. Este passo irá bloquear os logins baseados em senha. Por isso, garantir que você ainda terá acesso de administrador será essencial.

Assim que as condições acima forem verdadeiras, entre no seu servidor remoto com chaves SSH como root ou com uma conta com privilégios `sudo`. Abra o arquivo de configuração do daemon do SSH:

```
root@kali:/home/kali# vi /etc/ssh/sshd_config
```

Dentro do arquivo, procure por uma diretiva chamada `PasswordAuthentication`. Isso pode ser

transformado em comentário. Descomente a linha e configure o valor em no. Isso irá desativar a sua capacidade de fazer login via SSH usando senhas de conta:

```
PasswordAuthentication no
```

Salve e feche o arquivo quando você terminar. Para realmente implementar as alterações que acabamos de fazer, reinicie o serviço.

Em máquinas Ubuntu ou Debian, emita este comando:

```
sudo service ssh restart
```

Em máquinas RHEL, o daemon chama-se sshd:

```
sudo service sshd restart
```

Após completar este passo, você alterou seu daemon do SSH com sucesso para responder apenas a chaves SSH.# Definir modos de aplicação e permissivos para SELinux ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Estes foram agrupados em várias categorias.

- **Gerenciar segurança**
  - Defina as configurações do firewall usando firewall-cmd/firewalld
  - Criar e usar listas de controle de acesso a arquivos
  - Configure a autenticação baseada em chave para SSH
  - **Definir modos de aplicação e permissivos para SELinux**
  - Listar e identificar contexto de processos e arquivos do SELinux
  - Restaurar contextos de arquivo padrão
  - Use as configurações booleanas para modificar as configurações do SELinux do sistema
  - Diagnosticar e resolver violações de rotina da política do SELinux

Introdução

Security Enhanced Linux ou SELinux é um mecanismo de controle de acesso avançado integrado à maioria das distribuições Linux modernas. Ele foi inicialmente desenvolvido pela Agência de Segurança Nacional dos Estados Unidos para proteger os sistemas de computador

contra intrusões e adulterações maliciosas. Com o tempo, o SELinux foi lançado em domínio público e várias distribuições o incorporaram em seu código.

Muitos administradores de sistema consideram o SELinux um território um tanto desconhecido. O assunto pode parecer assustador e às vezes bastante confuso. No entanto, um sistema SELinux configurado corretamente pode reduzir muito os riscos de segurança e saber um pouco sobre isso pode ajudá-lo a solucionar mensagens de erro relacionadas ao acesso. Aqui, aprenderemos sobre os conceitos por trás do SELinux - seus pacotes, comandos e arquivos de configuração - e as mensagens de erro que ele registra quando o acesso é negado.

Por que SELinux

Antes de começar, vamos entender alguns conceitos.

O SELinux implementa o que é conhecido como **MAC** (Controle de Acesso Obrigatório). Isso é implementado em cima do que já está presente em todas as distribuições do Linux, o DAC (Controle de Acesso Discricionário).

Para entender o DAC, vamos primeiro considerar como funciona a segurança de arquivos tradicional do Linux.

Em um modelo de segurança tradicional, temos três entidades: Usuário, Grupo e Outro (u, g, o) que podem ter uma combinação de permissões de Leitura, Gravação e Execução (r, w, x) em um arquivo ou diretório. Se um usuário joatham criar um arquivo em seu diretório pessoal, esse usuário terá acesso de leitura / gravação a ele, assim como o grupo joatham. A "outra" entidade possivelmente não terá acesso a ela. No bloco de código a seguir, podemos considerar o conteúdo hipotético do diretório inicial de joatham.

Executando um comando como este:

```
[root@localhost joatham]# ls -l /home/joatham/
```

Agora joatham pode alterar este acesso. joatham pode conceder (e restringir) o acesso a este arquivo para outros usuários e grupos ou alterar o proprietário do arquivo. Essas ações podem deixar arquivos críticos expostos a contas que não precisam desse acesso. joatham também pode restringir para ser mais seguro, mas isso é discricionário: não há como o administrador do sistema aplicá-lo a todos os arquivos do sistema.

Considere outro caso: quando um processo Linux é executado, ele pode ser executado como usuário root ou outra conta com privilégios de superusuário. Isso significa que, se um hacker

assumir o controle do aplicativo, ele poderá usar esse aplicativo para obter acesso a qualquer recurso ao qual a conta do usuário tenha acesso. Para processos executados como usuário root, basicamente, isso significa tudo no servidor Linux.

Pense em um cenário em que você deseja restringir os usuários de executar scripts de shell de seus diretórios pessoais. Isso pode acontecer quando você tem desenvolvedores trabalhando em um sistema de produção. Você gostaria que eles vissem os arquivos de log, mas não quer que eles usem comandos `su` ou `sudo`, e não quer que eles executem nenhum script de seus diretórios pessoais. Como fazer isso?

O `SELinux` é uma maneira de ajustar esses requisitos de controle de acesso. Com ele, você pode definir o que um usuário ou processo pode fazer. Ele confina cada processo ao seu próprio domínio, de forma que o processo possa interagir apenas com certos tipos de arquivos e outros processos de domínios permitidos. Isso evita que um hacker sequestre qualquer processo para obter acesso de todo o sistema.

Configurando um Sistema de Teste

Para nos ajudar a aprender os conceitos, construiremos um servidor de teste rodando tanto um servidor web quanto um SFTP. No entanto, não configuraremos nenhum desses aplicativos.

Também criaremos algumas contas de usuário de teste em nosso servidor.

Finalmente, instalaremos os pacotes necessários relacionados ao `SELinux`. Isso é para garantir que possamos trabalhar com os comandos `SELinux` mais recentes.

Instalando Apache e SFTP Services

Primeiro, vamos fazer login no servidor como usuário **root** e executar o seguinte comando para instalar o Apache:

```
[root@localhost joatham]# yum install httpd
```

A saída mostrará o pacote sendo baixado e pedirá permissão para instalar:

```
[root@localhost joatham]# yum install httpd
Updating Subscription Management repositories.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)

     10 kB/s | 4.5 kB     00:00
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
```

```
     7.3 MB/s |  33 MB     00:04
Última verificação de data de vencimento de metadados: 0:00:26 atrás em qui 04 nov 2021
     14:29:03 -03.
Dependências resolvidas.
================================================================================

 Pacote                                  Arquitetura                  Versão
                                                                   Repositório
                                                Tamanho
================================================================================

Instalando:
 httpd                                   x86_64                       2.4.37-39.
      module+el8.4.0+12865+a7065a39.1                                rhel-8-for-x86_64-
      appstream-rpms                      1.4 M
Instalando dependências:
 apr                                     x86_64                       1.6.3-11.el8
                                                                   rhel-8-for-x86_64-appstream-
      rpms                              125 k
 apr-util                                x86_64                       1.6.1-6.el8
                                                                   rhel-8-for-x86_64-appstream
      -rpms                              105 k
 httpd-filesystem                        noarch                       2.4.37-39.
      module+el8.4.0+12865+a7065a39.1                                rhel-8-for-x86_64-
      appstream-rpms                      39 k
 httpd-tools                             x86_64                       2.4.37-39.
      module+el8.4.0+12865+a7065a39.1                                rhel-8-for-x86_64-
      appstream-rpms                      106 k
 mod_http2                               x86_64                       1.15.7-3.
      module+el8.4.0+8625+d397f3da                                   rhel-8-for-x86_64-
      appstream-rpms                      154 k
 redhat-logos-httpd                      noarch                       84.4-1.el8
                                                                    rhel-8-for-x86_64-baseos-
      rpms                                29 k
Instalando dependências fracas:
 apr-util-bdb                            x86_64                       1.6.1-6.el8
                                                                   rhel-8-for-x86_64-appstream
      -rpms                              25 k
 apr-util-openssl                        x86_64                       1.6.1-6.el8
                                                                   rhel-8-for-x86_64-appstream
      -rpms                              27 k
Ativando Fluxos de Módulos:
 httpd                                                                2.4

Resumo da transação
================================================================================

Instalar  9 Pacotes

Tamanho total do download: 2.0 M
Tamanho depois de instalado: 5.4 M
Correto? [s/N]: s
```

```
[root@localhost joatham]# systemctl start httpd
[root@localhost joatham]# systemctl status httpd●
 httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled
        )
   Active: active (running) since Thu 2021-11-04 14:33:05 -03; 7s ago
```

```
     Docs: man:httpd.service(8)
 Main PID: 4770 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 213 (limit: 17240)
   Memory: 24.7M
   CGroup: /system.slice/httpd.service
           |--4770 /usr/sbin/httpd -DFOREGROUND
           |--4775 /usr/sbin/httpd -DFOREGROUND
           |--4776 /usr/sbin/httpd -DFOREGROUND
           |--4777 /usr/sbin/httpd -DFOREGROUND
           `-4778 /usr/sbin/httpd -DFOREGROUND

nov 04 14:33:04 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
nov 04 14:33:05 localhost.localdomain httpd[4770]: AH00558: httpd: Could not reliably
     determine the server's fully qualified domain name, using localhost.localdomain. Set
     the 'ServerName' directive globally to >
nov 04 14:33:05 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
nov 04 14:33:05 localhost.localdomain httpd[4770]: Server configured, listening on: port 80
```

Em seguida, instalaremos o `vsftp`:

```
[root@localhost joatham]# systemctl start httpd
[root@localhost joatham]# systemctl status httpd●
 httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled
       )
   Active: active (running) since Thu 2021-11-04 14:33:05 -03; 7s ago
     Docs: man:httpd.service(8)
 Main PID: 4770 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 213 (limit: 17240)
   Memory: 24.7M
   CGroup: /system.slice/httpd.service
           |--4770 /usr/sbin/httpd -DFOREGROUND
           |--4775 /usr/sbin/httpd -DFOREGROUND
           |--4776 /usr/sbin/httpd -DFOREGROUND
           |--4777 /usr/sbin/httpd -DFOREGROUND
           `-4778 /usr/sbin/httpd -DFOREGROUND

nov 04 14:33:04 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
nov 04 14:33:05 localhost.localdomain httpd[4770]: AH00558: httpd: Could not reliably
     determine the server's fully qualified domain name, using localhost.localdomain. Set
     the 'ServerName' directive globally to >
nov 04 14:33:05 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
nov 04 14:33:05 localhost.localdomain httpd[4770]: Server configured, listening on: port 80
```

A seguir, usaremos o comando `systemctl start vsftpd` para iniciar o daemon `vsftpd`. A saída deve mostrar algo como o seguinte:

```
[root@localhost joatham]# systemctl start vsftpd
[root@localhost joatham]# systemctl status vsftpd●
 vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; vendor preset:
```

```
        disabled)
   Active: active (running) since Thu 2021-11-04 14:36:40 -03; 7s ago
  Process: 5167 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/
       SUCCESS)
 Main PID: 5168 (vsftpd)
    Tasks: 1 (limit: 17240)
   Memory: 576.0K
   CGroup: /system.slice/vsftpd.service
           `-5168 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

nov 04 14:36:39 localhost.localdomain systemd[1]: Starting Vsftpd ftp daemon...
nov 04 14:36:40 localhost.localdomain systemd[1]: Started Vsftpd ftp daemon.
```

Instalando Pacotes SELinux

Vários pacotes são usados no SELinux. Alguns são instalados por padrão. Aqui está uma lista para distribuições baseadas no Red Hat:

- **policycoreutils** (fornece utilitários para gerenciar SELinux).
- **policycoreutils-python** (fornece utilitários para gerenciar SELinux).
- **selinux-policy** (fornece a política de referência do SELinux).
- **selinux-policy-oriented** (fornece política de direcionamento SELinux).
- **libselinux-utils** (fornece algumas ferramentas para gerenciar SELinux).
- **setroubleshoot-server** (fornece ferramentas para decifrar mensagens de registro de auditoria).
- **setools** (fornece ferramentas para monitoramento de registro de auditoria, política de consulta e gerenciamento de contexto de arquivo).
- **setools-console** (fornece ferramentas para monitoramento de registro de auditoria, política de consulta e gerenciamento de contexto de arquivo).
- **mcstrans** (ferramentas para traduzir diferentes níveis para um formato fácil de entender)

Alguns deles já estão instalados. Verifique:

A saída deve ser semelhante a esta:

```
[root@localhost joatham]# rpm -qa | grep selinux
python3-libselinux-2.9-5.el8.x86_64
container-selinux-2.167.0-1.module+el8.4.0+12646+b6fd1bdf.noarch
rpm-plugin-selinux-4.14.3-14.el8_4.x86_64
selinux-policy-3.14.3-67.el8_4.2.noarch
flatpak-selinux-1.8.5-3.el8.noarch
libselinux-utils-2.9-5.el8.x86_64
selinux-policy-targeted-3.14.3-67.el8_4.2.noarch
libselinux-2.9-5.el8.x86_64
```

Você pode ir em frente e instalar todos os pacotes com o comando abaixo (o yum irá apenas atualizar qualquer um que você já tenha), ou apenas aqueles que você achar que faltam em seu sistema:

```
[root@localhost joatham]# yum install policycoreutils policycoreutils-python selinux-policy
      selinux-policy-targeted libselinux-utils setroubleshoot-server setools setools-
     console mcstrans
```

Agora devemos ter um sistema carregado com todos os pacotes SELinux. Também temos servidores Apache e SFTP em execução com suas configurações padrão.

Modos SELinux

É hora de começar a brincar com o SELinux, então vamos começar com os modos do SELinux. A qualquer momento, o SELinux pode estar em qualquer um dos três modos possíveis:

- **Enforcing**
- **Permissive**
- **Disabled**

No modo de imposição (Enforcing), o SELinux aplicará sua política no sistema Linux e garantirá que todas as tentativas de acesso não autorizado por usuários e processos sejam negadas. As negações de acesso também são gravadas em arquivos de log relevantes. Falaremos sobre as políticas do SELinux e logs de auditoria mais tarde.

O modo permissivo é como um estado semi-habilitado. O SELinux não aplica sua política no modo permissivo, portanto, nenhum acesso é negado. No entanto, qualquer violação de política ainda é registrada nos registros de auditoria. É uma ótima maneira de testar o SELinux antes de aplicá-lo.

O modo desabilitado é autoexplicativo - o sistema não funcionará com segurança aprimorada.

Verificando os modos e status do SELinux

Podemos executar o comando getenforce para verificar o modo SELinux atual.

O SELinux deve estar no modo *enforcing* no momento, então a saída ficará assim:

```
[root@localhost joatham]# getenforce
Enforcing
```

Também podemos executar o comando `sestatus`:

```
[root@localhost joatham]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

Arquivo de configuração SELinux

O arquivo de configuração principal do SELinux é `/etc/selinux/config`. Podemos executar o seguinte comando para visualizar seu conteúdo:

```
[root@localhost joatham]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Existem duas diretivas neste arquivo. A diretiva **SELINUX** determina o modo `SELinux` e pode ter três valores possíveis, conforme discutimos antes.

A diretiva **SELINUXTYPE** determina a política que será usada. O valor padrão é `targeted`. Com uma política direcionada, o SELinux permite que você personalize e ajuste as permissões de controle de acesso. O outro valor possível é `MLS` (segurança multinível), um modo avançado de proteção. Também com o `MLS`, você precisa instalar um pacote adicional.

Habilitando e desabilitando o SELinux

Habilitar o SELinux é bastante simples; mas, ao contrário de desativá-lo, deve ser feito em um processo de duas etapas. Assumimos que o SELinux está *enforcing* e que você instalou todos os pacotes do SELinux.

Como primeiro passo, precisamos editar o arquivo `/etc/selinux/config` para alterar a diretiva SELINUX para o modo permissivo.

```
[root@localhost joatham]# vi /etc/selinux/config
```

```
...
SELINUX=permissive
...
```

Definir o status como **permissive** primeiro é necessário porque cada arquivo no sistema precisa ter seu contexto rotulado antes que o SELinux possa ser executado. A menos que todos os arquivos estejam devidamente rotulados, os processos em execução em domínios confinados podem falhar porque eles não podem acessar os arquivos com os contextos corretos. Isso pode fazer com que o processo de inicialização falhe ou comece com erros. Apresentaremos contextos e domínios posteriormente no tutorial.

Agora reinicie o sistema:

```
[root@localhost joatham]# reboot
```

O processo de reinicialização verá todos os arquivos no servidor rotulados com um contexto SELinux. Como o sistema está funcionando em modo permissivo, erros SELinux e negações de acesso serão relatados, mas nada parará.

Faça login no seu servidor novamente como **root**. Em seguida, pesquise a string "SELinux está impedindo" no conteúdo do arquivo /var/log/messages.

```
[root@localhost joatham]# cat /var/log/messages | grep "SELinux is preventing"
```

```
[root@localhost joatham]# cat /var/log/messages | grep "SELinux"
```

Volte seu ambiente para um estilo mais seguro, edite novamente para enforcing no arquivo de configuracao `/etc/sysconfig/selinux`:

```
...
```

```
SELINUX=enforcing
...
```

Em seguida, reinicie o servidor novamente.

```
[root@localhost joatham]# reboot
```

Assim que o servidor estiver online novamente, podemos executar o comando `sestatus` para verificar o status do SELinux.

```
[root@localhost joatham]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

Também podemos alternar temporariamente entre os modos obrigatório e permissivo usando o comando `setenforce`. (Observe que não podemos executar `setenforce` quando o SELinux está desativado.)

Primeiro mude o modo `SELinux` de obrigatório para permissivo em nosso sistema:

```
[root@localhost joatham]# setenforce permissive
```

A execução do comando `sestatus` agora mostra que o modo atual é diferente do modo definido no arquivo de configuração:

```
[root@localhost joatham]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
```

```
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

Volte para a aplicação de enforcing:

```
[root@localhost joatham]# setenforce enforcing
```

> Da para alterar os modos atraves de numeros por ex: `setenforce 0`# Listar e identificar contexto de processos e arquivos do SELinux ## Pontos de estudo para o exame Os candidatos ao exame `RHCSA` devem ser capazes de realizar as tarefas abaixo sem assistência. Estes foram agrupados em várias categorias.

- **Gerenciar segurança**
  - Defina as configurações do firewall usando firewall-cmd/firewalld
  - Criar e usar listas de controle de acesso a arquivos
  - Configure a autenticação baseada em chave para SSH
  - Definir modos de aplicação e permissivos para SELinux
  - **Listar e identificar contexto de processos e arquivos do SELinux**
  - Restaurar contextos de arquivo padrão
  - Use as configurações booleanas para modificar as configurações do SELinux do sistema
  - Diagnosticar e resolver violações de rotina da política do SELinux

Introdução

Primeiro, vamos criar quatro contas de usuário para demonstrar os recursos do `SELinux` à medida que avançamos.

- usuario_regular
- switchuser
- usuario_convidado
- usuario_restrito

Você atualmente deve ser o usuário **root**. Vamos executar o seguinte comando para adicionar a conta de **usuario_regular**:

```
[root@localhost joatham]# useradd -c "Usuario regular" usuario_regular
```

Em seguida, executamos o comando `passwd` para alterar sua senha:

```
[root@localhost joatham]# passwd usuario_regular
```

Vamos criar as outras contas também:

```
[root@localhost joatham]# useradd -c "Switched User" switcheduser
[root@localhost joatham]# useradd -c "Switched User" switcheduser
```

```
[root@localhost joatham]# useradd -c "Usuario Convidado" usuario_convidado
[root@localhost joatham]# passwd usuario_convidado
```

```
[root@localhost joatham]# useradd -c "Usuario restrito" usuario_restrito
[root@localhost joatham]# passwd usuario_restrito
```

SELinux para processos e arquivos

O objetivo do SELinux é proteger como os processos acessam os arquivos em um ambiente Linux. Sem o SELinux, um processo ou aplicativo como o daemon do Apache será executado no contexto do usuário que o iniciou. Portanto, se o seu sistema for comprometido por um aplicativo nocivo que está sendo executado sob o usuário root, o aplicativo pode fazer o que quiser porque o root tem direitos abrangentes sobre todos os arquivos.

O SELinux tenta dar um passo adiante e eliminar esse risco. Com o SELinux, um processo ou aplicativo terá apenas os direitos necessários para funcionar e NADA mais. A política SELinux do aplicativo determinará a quais tipos de arquivos ele precisa acessar e para quais processos pode fazer a transição . As políticas SELinux são escritas por desenvolvedores de aplicativos e enviadas com a distribuição Linux que as suporta. Uma política é basicamente um conjunto de regras que mapeia processos e usuários de acordo com seus direitos.

A primeira parte da segurança coloca um rótulo em cada entidade no sistema Linux. Um rótulo é como qualquer outro arquivo ou atributo de processo (proprietário, grupo, data de criação, etc.); mostra o contexto do recurso. Então, o que é um contexto? Simplificando, um contexto é uma coleção de informações relacionadas à segurança que ajuda o SELinux a tomar decisões de controle de acesso. Tudo em um sistema Linux pode ter um contexto de segurança: uma conta de usuário, um arquivo, um diretório, um daemon ou uma porta podem

ter seus contextos de segurança. No entanto, o contexto de segurança vai significar coisas diferentes para diferentes tipos de objetos.

Contextos de arquivo SELinux

Vamos começar entendendo os contextos dos arquivos SELinux. Vejamos a saída de um comando `ls -l` regular no diretório `/etc`.

Isso nos mostrará uma saída familiar:

```
[root@localhost joatham]# ls -l /etc/*.conf
-rw-r--r--. 1 root root       55 fev  1  2021 /etc/asound.conf
-rw-r--r--. 1 root root    25696 out 21  2020 /etc/brltty.conf
-rw-r--r--. 1 root root     1083 mai 10  2019 /etc/chrony.conf
-rw-r--r--. 1 root root     1174 dez 10  2020 /etc/dleyna-server-service.conf
-rw-r--r--. 1 root dnsmasq 26843 dez 14  2020 /etc/dnsmasq.conf
-rw-r--r--. 1 root root      117 jan 21  2021 /etc/dracut.conf
```

Simples, certo? Vamos agora adicionar o sinalizador `-Z`:

```
[root@localhost joatham]# ls -Z /etc/*.conf
```

Agora temos uma coluna extra de informações após a propriedade do usuário e do grupo:

```
...
-rw-r--r--. 1 root root    system_u:object_r:etc_t:s0             55 fev  1  2021 /etc/
     asound.conf
-rw-r--r--. 1 root root    system_u:object_r:etc_t:s0          25696 out 21  2020 /etc/
     brltty.conf
-rw-r--r--. 1 root root    system_u:object_r:etc_t:s0           1083 mai 10  2019 /etc/
     chrony.conf
-rw-r--r--. 1 root root    system_u:object_r:etc_t:s0           1174 dez 10  2020 /etc/
     dleyna-server-service.conf
-rw-r--r--. 1 root dnsmasq system_u:object_r:dnsmasq_etc_t:s0  26843 dez 14  2020 /etc/
     dnsmasq.conf
-rw-r--r--. 1 root root    system_u:object_r:etc_t:s0            117 jan 21  2021 /etc/
     dracut.conf
...
```

Esta coluna mostra os contextos de segurança dos arquivos. Diz-se que um arquivo foi rotulado com seu contexto de segurança quando você tem essas informações disponíveis para ele. Vamos dar uma olhada mais de perto em um dos contextos de segurança.

```
-rw-r--r--. root    root  system_u:object_r:etc_t:s0       /etc/logrotate.conf
```

O contexto de segurança é esta parte:

```
system_u:object_r:etc_t:s0
```

Existem quatro partes e cada parte do contexto de segurança é separada por dois pontos (:). A primeira parte é o contexto do usuário SELinux para o arquivo. Discutiremos os usuários do SELinux mais tarde, mas por enquanto, podemos ver que é `system_u`. Cada conta de usuário do Linux é mapeada para um usuário SELinux e, neste caso, o usuário `root` que possui o arquivo é mapeado para o usuário SELinux `system_u`. Esse mapeamento é feito pela política SELinux.

A segunda parte especifica a função SELinux, que é `object_r` .

O que é mais importante aqui é a terceira parte, o tipo de arquivo listado aqui como `etc_t`. Esta é a parte que define a que tipo o arquivo ou diretório pertence. Podemos ver que a maioria dos arquivos pertence ao tipo `etc_t` no diretório `/etc`. Hipoteticamente, você pode pensar no tipo como uma espécie de "grupo" ou atributo do arquivo: é uma forma de classificar o arquivo.

Também podemos ver que alguns arquivos podem pertencer a outros tipos, como o `locale. conf` que tem um tipo `locale_t`. Mesmo quando todos os arquivos listados aqui têm o mesmo usuário e proprietários de grupo, seus tipos podem ser diferentes.

Como outro exemplo, vamos verificar os contextos de tipo para os diretórios iniciais do usuário:

```
ls -Z /home
```

Os diretórios pessoais terão um tipo de contexto diferente: `user_home_dir_t`

```
[root@localhost joatham]# ls -lZ /home/
total 4
drwx------.  4 cerpa             cerpa             unconfined_u:object_r:user_home_dir_t:s0
       113 nov  4 00:57 cerpa
drwx------.  4 hieneken          hieneken          unconfined_u:object_r:user_home_dir_t:s0
       113 nov  4 00:26 hieneken
drwx------. 16 joatham           joatham           unconfined_u:object_r:user_home_dir_t:s0
      4096 nov  4 15:02 joatham
drwx------.  3 pedro             pedro             unconfined_u:object_r:user_home_dir_t:s0
        78 nov  3 17:25 pedro
drwx------.  3 switcheduser      switcheduser      unconfined_u:object_r:user_home_dir_t:s0
```

```
        78 nov  4 15:38 switcheduser
drwx------.  4 usuario1          usuario1          unconfined_u:object_r:user_home_dir_t:s0
       129 nov  3 17:01 usuario1
drwx------.  4 usuario2          usuario2          unconfined_u:object_r:user_home_dir_t:s0
       113 nov  3 16:16 usuario2
drwx------.  3 usuario_convidado usuario_convidado unconfined_u:object_r:user_home_dir_t:s0
        78 nov  4 15:39 usuario_convidado
drwx------.  3 usuario_regular   usuario_regular   unconfined_u:object_r:user_home_dir_t:s0
        78 nov  4 15:37 usuario_regular
drwx------.  3 usuario_restrito  usuario_restrito  unconfined_u:object_r:user_home_dir_t:s0
        78 nov  4 15:39 usuario_restrito
```

A quarta parte do contexto de segurança, `s0`, tem a ver com segurança multinível ou MLS. Basicamente, esta é outra forma de impor a política de segurança do SELinux, e esta parte mostra a sensibilidade do recurso (`s0`). Falaremos brevemente sobre sensibilidade e categorias mais tarde. Para a maioria das configurações básicas do SELinux, os três primeiros contextos de segurança são mais importantes.

Contextos do processo SELinux

Vamos agora falar sobre os contextos de segurança do processo.

Inicie os serviços Apache e SFTP. Instalamos esses serviços no primeiro tutorial do SELinux.

```
[root@localhost joatham]# systemctl start httpd
[root@localhost joatham]# systemctl start vsftpd
```

Podemos executar o comando `ps` com alguns sinalizadores para mostrar os processos Apache e VFTP em execução em nosso servidor:

```
ps -efZ | grep 'httpd\|vsftpd'
```

Mais uma vez, o sinalizador `-Z` é usado para exibir contextos SELinux. A saída mostra o usuário executando o processo, o ID do processo e o ID do processo pai:

```
[root@localhost joatham]# ps -efZ | grep 'httpd\|vsftpd'
system_u:system_r:httpd_t:s0    root        3328       1  0 15:16 ?        00:00:00 /usr/
     sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache      3333    3328  0 15:16 ?        00:00:00 /usr/
     sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache      3334    3328  0 15:16 ?        00:00:00 /usr/
     sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache      3335    3328  0 15:16 ?        00:00:00 /usr/
```

```
      sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0    apache      3336    3328  0 15:16 ?        00:00:00 /usr/
      sbin/httpd -DFOREGROUND
system_u:system_r:ftpd_t:s0-s0:c0.c1023 root 4032      1  0 15:53 ?        00:00:00 /usr/
      sbin/vsftpd /etc/vsftpd/vsftpd.conf
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 4041 3055  0 15:53 pts/0
      00:00:00 grep --color=auto httpd\|vsftpd
```

O contexto de segurança é esta parte:

```
system_u:system_r:httpd_t:s0
```

O contexto de segurança tem quatro partes: **usuário**, **função**, **domínio** e **sensibilidade**. O usuário, a função e a sensibilidade funcionam da mesma forma que os mesmos contextos para arquivos. O domínio é exclusivo para processos.

No exemplo acima, podemos ver que alguns processos estão sendo executados no domínio `httpd_t`, enquanto um está sendo executado no domínio `ftpd_t`.

Então, o que o domínio está fazendo pelos processos? Dá ao processo um contexto para ser executado. É como uma bolha em torno do processo que o confina. Diz ao processo o que pode e o que não pode fazer. Este confinamento garante que cada domínio de processo possa atuar em apenas certos tipos de arquivos e nada mais.

Usando esse modelo, mesmo se um processo for sequestrado por outro processo ou usuário malicioso, o pior que pode fazer é danificar os arquivos aos quais tem acesso. Por exemplo, o daemon vsftp não terá acesso aos arquivos usados por, digamos, sendmail ou samba. Essa restrição é implementada a partir do nível do kernel: é aplicada conforme a política SELinux carrega na memória e, portanto, o controle de acesso se torna obrigatório .

Convenções de Nomenclatura

Antes de prosseguirmos, aqui está uma observação sobre a convenção de nomenclatura SELinux. Os usuários do SELinux são sufixados por `_u`, as funções são sufixadas por `_r` e os tipos (para arquivos) ou domínios (para processos) são sufixados por `_t`.

Como os processos acessam os recursos

Até agora, vimos que arquivos e processos podem ter contextos diferentes e que estão restritos a seus próprios tipos ou domínios. Então, como funciona um processo? Para ser executado, um processo precisa acessar seus arquivos e realizar algumas ações neles (abrir, ler, modificar

ou executar). Também aprendemos que cada processo pode ter acesso a apenas certos tipos de recursos (arquivos, diretórios, portas, etc.).

O SELinux estipula essas regras de acesso em uma política. As regras de acesso seguem uma estrutura de declaração de permissão padrão:

```
allow <domain> <type>:<class> { <permissions> };
```

Já falamos sobre domínios e tipos. A **classe** define o que o recurso realmente representa (arquivo, diretório, link simbólico, dispositivo, portas, cursor etc.)

Aqui está o que significa esta declaração genérica de permissão:

- Se um processo é de determinado domínio
- E o objeto de recurso que está tentando acessar é de determinada classe e tipo
- Em seguida, permita o acesso
- Caso contrário, negue o acesso

Para ver como isso funciona, vamos considerar os contextos de segurança do daemon httpd em execução em nosso sistema:

```
system_u:system_r:httpd_t:s0     7126 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0     7127 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0     7128 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0     7129 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0     7130 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0     7131 ?        00:00:00 httpd
```

O diretório inicial padrão para o servidor da web é `/var/www/html`. Vamos criar um arquivo dentro desse diretório e verificar seu contexto:

```
[root@localhost html]# touch index.html
[root@localhost html]# ls -lZ
```

O contexto do arquivo para nosso conteúdo da web será `httpd_sys_content_t` :

```
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/index.html
```

Podemos usar o comando `sesearch` para verificar o tipo de acesso permitido para o daemon `httpd`:

“shell [root@localhost html]# yum install setools-console “‘

```
[root@localhost html]# sesearch --allow --source httpd_t --target httpd_sys_content_t --
     class file
```

As flags usadas com o comando são bastante autoexplicativos: o domínio de origem é `httpd_t`, o mesmo domínio no qual o Apache está sendo executado. Estamos interessados em recursos de destino que são arquivos e têm um contexto de tipo de `httpd_sys_content_t`. Sua saída deve ser assim:

```
allow domain file_type:file map; [ domain_can_mmap_files ]:True
allow httpd_t httpd_content_type:file { getattr ioctl lock map open read };
allow httpd_t httpd_content_type:file { getattr ioctl lock open read }; [
     httpd_builtin_scripting ]:True
allow httpd_t httpd_sys_content_t:file { getattr ioctl lock map open read };
allow httpd_t httpdcontent:file { append create getattr ioctl link lock open read rename
     setattr unlink write }; [ ( httpd_builtin_scripting && httpd_unified &&
     httpd_enable_cgi ) ]:True
allow httpd_t httpdcontent:file { execute execute_no_trans getattr ioctl map open read }; [
      ( httpd_builtin_scripting && httpd_unified && httpd_enable_cgi ) ]:True
```

Observe a primeira linha:

```
allow httpd_t httpd_sys_content_t : file { ioctl read getattr lock open } ;
```

Isso diz que o daemon httpd (o servidor da Web Apache) tem controle de E/S, leitura, obtenção de atributos, bloqueio e acesso aberto a arquivos do tipo httpd_sys_content. Neste caso, nosso arquivo `index.html` é do mesmo tipo.

Indo um passo adiante, vamos primeiro modificar a página da web (/var/www/html/index.html). Edite o arquivo para conter este conteúdo:

```
<html>
    <title>
        4LINUX CABECALHO
    </title>
    <body>
        <h1>PAGINA DE TESTE</h1>
```

```
    </body>
</html>
```

Em seguida, vamos alterar a permissão da pasta `/var/www/` e seu conteúdo, seguido por uma reinicialização do daemon `httpd`:

```
[root@localhost var]# chmod -R 755 /var/www/
[root@localhost var]# systemctl restart httpd
```

Em seguida, tentaremos acessá-lo de um navegador:

Acessando a página da Web com a configuração SELinux correta

Até agora tudo bem. O daemon `httpd` está autorizado a acessar um determinado tipo de arquivo e podemos vê-lo ao acessar através do navegador. A seguir, vamos tornar as coisas um pouco diferentes alterando o contexto do arquivo. Usaremos o comando `chcon` para isso. A flag `--type` para o comando nos permite especificar um novo tipo para o recurso de destino. Aqui, estamos mudando o tipo de arquivo para `var_t`.

```
[root@localhost var]# chcon --type var_t /var/www/html/index.html
```

Podemos confirmar a mudança de tipo:

```
[root@localhost var]# ls -lZ /var/www/html/
```

```
-rwxr-xr-x. 1 root root unconfined_u:object_r:var_t:s0 122 nov  4 16:15 index.html
```

Em seguida, quando tentamos acessar a página da web (ou seja, o daemon httpd tenta ler o arquivo), você pode obter um erro **Proibido** ou pode ver a página genérica.

Então, o que está acontecendo aqui? Obviamente, algum acesso agora está sendo negado, mas de quem é esse acesso? No que diz respeito ao SELinux, o servidor web está autorizado a acessar apenas certos tipos de arquivos e `var_t` não é um desses contextos. Como alteramos o contexto do arquivo index.html para `var_t`, o Apache não pode mais lê-lo e obtemos um erro.

Para fazer as coisas funcionarem novamente, vamos mudar o tipo de arquivo com o comando

restorecon. A opção -v mostra a mudança dos rótulos de contexto:

```
[root@localhost var]# restorecon -v /var/www/html/index.html
```

```
Relabeled /var/www/html/index.html from unconfined_u:object_r:var_t:s0 to unconfined_u:
     object_r:httpd_sys_content_t:s0
```

Se tentarmos acessar a página agora, ela mostrará nosso texto novamente.

Este é um conceito importante a ser entendido: certificar-se de que os arquivos e diretórios tenham o contexto correto é fundamental para garantir que o SELinux esteja se comportando como deveria. # Restaurar contextos de arquivo padrão ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Estes foram agrupados em várias categorias.

- **Gerenciar segurança**
  - Defina as configurações do firewall usando firewall-cmd/firewalld
  - Criar e usar listas de controle de acesso a arquivos
  - Configure a autenticação baseada em chave para SSH
  - Definir modos de aplicação e permissivos para SELinux
  - Listar e identificar contexto de processos e arquivos do SELinux
  - **Restaurar contextos de arquivo padrão**
  - Use as configurações booleanas para modificar as configurações do SELinux do sistema
  - Diagnosticar e resolver violações de rotina da política do SELinux

Introdução

O SELinux impõe algo que podemos denominar como "herança de contexto". O que isso significa é que, a menos que especificado pela política, os processos e arquivos são criados com os contextos de seus pais.

Portanto, se tivermos um processo denominado "proc_a" gerando outro processo denominado "proc_b", o processo gerado será executado no mesmo domínio que "proc_a", a menos que especificado de outra forma pela política SELinux.

Da mesma forma, se tivermos um diretório com um tipo de "algum_contexto_t", qualquer arquivo ou diretório criado sob ele terá o mesmo tipo de contexto, a menos que a política diga o contrário.

Para ilustrar isso, vamos verificar os contextos do diretorio `/var/www/`

```
[root@localhost var]# ls -lZ
```

O diretório `html` dentro `/var/www/` tem o contexto do tipo `httpd_sys_content_t`. Como vimos antes, o arquivo `index.html` dentro dele tem o mesmo contexto (ou seja, o contexto do pai):

```
[root@localhost var]# ls -lZ www/
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0  6 out  6 13:46 cgi-
     bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0     24 nov  4 16:15 html
```

Essa herança não é preservada quando os arquivos são copiados para outro local. Em uma operação de cópia, o arquivo ou diretório copiado assumirá o contexto de tipo do local de destino. No comando abaixo, estamos copiando o arquivo `index.html` (com o contexto do tipo "httpd_sys_content_t") para o diretório `/var/`:

```
[root@localhost var]# cp /var/www/html/index.html /var/
```

Se verificarmos o contexto do arquivo copiado, veremos que ele mudou para `var_t`, o contexto de seu diretório pai atual:

```
[root@localhost var]# ls -lZ index.html
```

```
-rwxr-xr-x. 1 root root unconfined_u:object_r:var_t:s0 122 nov  4 16:27 index.html
```

Esta mudança de contexto pode ser substituída pela cláusula `--preserver=context` no comando `cp`.

Quando arquivos ou diretórios são movidos, os contextos originais são preservados. No comando a seguir, estamos movendo o `/var/index.html` para o diretório `/etc/`:

```
[root@localhost var]# mv index.html /etc/
```

Quando verificamos o contexto do arquivo movido, vemos que o contexto `var_t` foi preservado no diretório `/etc/`:

```
[root@localhost ~]# ls -lZ /etc/index.html
-rwxr-xr-x. 1 root root unconfined_u:object_r:var_t:s0 122 nov  4 16:27 /etc/index.html
```

Então, por que estamos tão preocupados com os contextos dos arquivos? Por que esse conceito de copiar e mover é importante? Pense nisso: talvez você tenha decidido copiar todos os arquivos HTML do seu servidor web para um diretório separado na pasta raiz. Você fez isso para simplificar o processo de backup e também para aumentar a segurança: você não quer que nenhum hacker adivinhe facilmente onde estão os arquivos do seu site. Você atualizou o controle de acesso do diretório, alterou o arquivo de configuração da web para apontar para o novo local, reiniciou o serviço, mas ainda não funciona. Talvez você possa examinar os contextos do diretório e seus arquivos como a próxima etapa de solução de problemas. Vamos ver como um exemplo prático.

SELinux em ação: testando um erro de contexto de arquivo

Primeiro, vamos criar um diretório na raiz denominado `www`. Também criaremos uma pasta chamada `html` abaixo `www`.

```
[root@localhost ~]# mkdir -p /www/html
```

Se executarmos o comando `ls -Z`, veremos que esses diretórios foram criados com o contexto `default_t`:

```
[root@localhost ~]# ls -lZ /www/
total 0
drwxr-xr-x. 2 root root unconfined_u:object_r:default_t:s0 6 nov  4 16:30 html
```

Em seguida, copiamos o conteúdo do diretório `/var/www/html` para `/www/html`:

```
[root@localhost ~]# cp /var/www/html/index.html /www/html/
```

O arquivo copiado terá um contexto de `default_t`. Esse é o contexto do diretório pai.

Agora, editamos o arquivo `httpd.conf` para apontar para esse novo diretório como a pasta raiz do site.

```
[root@localhost ~]# vi +122 /etc/httpd/conf/httpd.conf
```

Primeiro, comentamos a localização existente para a raiz do documento e adicionamos uma nova diretiva `DocumentRoot` para `/www/html`:

```
# DocumentRoot "/var/www/html"

DocumentRoot "/www/html"
```

Também comentamos a seção de direitos de acesso para a raiz do documento existente e adicionamos uma nova seção:

```
#<Directory "/var/www">
#    AllowOverride None
    # Allow open access:
#    Require all granted
#</Directory>

<Directory "/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>
```

Deixamos a localização do diretório `cgi-bin` como está. Não estamos entrando na configuração detalhada do Apache aqui; queremos apenas que nosso site funcione para os propósitos do SELinux.

Finalmente, reinicie o daemon httpd:

```
[root@localhost ~]# systemctl restart httpd
```

Assim que o servidor for reiniciado, acessar a página da web nos dará o mesmo erro que vimos antes.

O erro está acontecendo porque o contexto do arquivo `index.html` mudou durante a operação de cópia. Ele precisa ser alterado de volta ao seu contexto original (httpd_sys_content_t).

Mas como nós fazemos isso?

Alterando e restaurando contextos de arquivo SELinux

Em um exemplo de código anterior, vimos dois comandos para alterar o conteúdo do arquivo: `chcon` e `restorecon`. rodar `chcon` é uma medida temporária. Você pode usá-lo para alterar temporariamente os contextos de arquivo ou diretório para solucionar erros de negação de acesso. No entanto, esse método é apenas temporário: uma renomeação do sistema de arquivos ou a execução do comando `restorecon` reverterá o arquivo de volta ao seu contexto original.

Além disso, a execução exige `chcon` que você conheça o contexto correto para o arquivo; a flag `--type` especifica o contexto do destino. `restorecon` não precisa disso especificado. Se você executar `restorecon`, o arquivo terá o contexto correto reaplicado e as alterações se tornarão permanentes.

Mas se você não souber o contexto correto do arquivo, como o sistema saberá qual contexto aplicar quando for executado `restorecon`?

Convenientemente, o SELinux "lembra" o contexto de cada arquivo ou diretório no servidor. No RHEL, os contextos de arquivos já existentes no sistema são listados no arquivo `/etc/selinux/targeted/contexts/files/file_contexts`. É um arquivo grande e lista todos os tipos de arquivos associados a todos os aplicativos suportados pela distribuição Linux. Contextos de novos diretórios e arquivos são registrados no arquivo `/etc/selinux/targeted/contexts/files/file_contexts.local`. Portanto, quando executamos o comando `restorecon`, o SELinux procura o contexto correto de um desses dois arquivos e o aplica ao destino.

O comando abaixo mostra um trecho de um dos arquivos:

```
[root@localhost ~]# cat /etc/selinux/targeted/contexts/files/file_contexts | less
```

```
...
/bin/ksh.*      --      system_u:object_r:shell_exec_t:s0
/bin/zsh.*      --      system_u:object_r:shell_exec_t:s0
/dev/md/.*      --      system_u:object_r:mdadm_var_run_t:s0
/dev/adb.*      -c      system_u:object_r:tty_device_t:s0
/dev/bsr.*      -c      system_u:object_r:cpu_device_t:s0
/dev/cmx.*      -c      system_u:object_r:smartcard_device_t:s0
/dev/cpu.*      -c      system_u:object_r:cpu_device_t:s0
/dev/dlm.*      -c      system_u:object_r:dlm_control_device_t:s0
/dev/dsp.*      -c      system_u:object_r:sound_device_t:s0
/dev/hvc.*      -c      system_u:object_r:tty_device_t:s0

...
```

Para alterar permanentemente o contexto do nosso arquivo `index.html` em `/www/html`, temos que seguir um processo de duas etapas.

Primeiro, executamos o comando `semanage fcontext`. Isso gravará o novo contexto no arquivo `/etc/selinux/targeted/contexts/files/file_contexts.local`. Mas não vai renomear o próprio arquivo. Faremos isso para os dois diretórios.

```
semanage fcontext --add --type httpd_sys_content_t "/www(/.*)?"
semanage fcontext --add --type httpd_sys_content_t "/www/html(/.*)?"
```

Para ter certeza, podemos verificar o banco de dados de contexto do arquivo (observe que estamos usando o arquivo file_contexts.local):

```
[root@localhost ~]# cat /etc/selinux/targeted/contexts/files/file_contexts.local
# This file is auto-generated by libsemanage
# Do not edit directly.

/opt/VBoxGuestAdditions-6.1.18/other/mount.vboxsf    system_u:object_r:mount_exec_t:s0
/usr/bin/VBoxClient    system_u:object_r:bin_t:s0
/www(/.*)?    system_u:object_r:httpd_sys_content_t:s0
/www/html(/.*)?    system_u:object_r:httpd_sys_content_t:s0
```

A seguir, executaremos o comando `restorecon`. Isso irá rotular novamente o arquivo ou diretório com o que foi registrado na etapa anterior:

```
[root@localhost ~]# restorecon -Rv /www/
```

Isso deve redefinir o contexto em três níveis: o diretório `/www` de nível superior, o diretório `/www/html` abaixo dele e o arquivo `index.html` em `/www/html`:

```
Relabeled /www from unconfined_u:object_r:default_t:s0 to unconfined_u:object_r:
     httpd_sys_content_t:s0
Relabeled /www/html from unconfined_u:object_r:default_t:s0 to unconfined_u:object_r:
     httpd_sys_content_t:s0
Relabeled /www/html/index.html from unconfined_u:object_r:default_t:s0 to unconfined_u:
     object_r:httpd_sys_content_t:s0
```

Se tentarmos acessar a página da web, ela deve funcionar.

Existe uma ferramenta bacana chamada `matchpathcon` que pode ajudar a solucionar problemas relacionados ao contexto. Este comando examinará o contexto atual de um recurso e o comparará com o que está listado no banco de dados de contexto SELinux. Se for diferente, irá sugerir a mudança necessária. Vamos testar isso com o arquivo `/www/html/index.html`. Usaremos a flag `-V` que verifica o contexto:

```
[root@localhost ~]# matchpathcon -V /www/html/index.html
```

A saída `matchpathcon` deve mostrar que o contexto foi verificado.

```
/www/html/index.html verified.
```

Para um arquivo rotulado incorretamente, a mensagem dirá qual deve ser o contexto:

`shell /www/html/index.html has context unconfined_u:object_r:default_t:s0, should be system_u:object_r :httpd_sys_content_t:s0#` Use as configurações booleanas para modificar as configurações do SELinux do sistema ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar segurança**
  - Defina as configurações do firewall usando firewall-cmd/firewalld
  - Crie e use listas de controle de acesso a arquivos
  - Configure a autenticação baseada em chave para SSH
  - Defina modos de aplicação e permissivos para SELinux
  - Liste e identifique o contexto de processos e arquivos do SELinux
  - Restaure contextos de arquivo padrão
  - **Use as configurações booleanas para modificar as configurações do SELinux do sistema**
  - Diagnostique e resolva violações de rotina da política do SELinux

Comportamento da política SELinux

A política SELinux não é algo que substitui a segurança DAC tradicional. Se uma regra DAC proíbe o acesso de um usuário a um arquivo, as regras de política SELinux não serão avaliadas porque a primeira linha de defesa já bloqueou o acesso. As decisões de segurança SELinux entram em jogo depois que a segurança DAC foi avaliada.

Quando um sistema habilitado para SELinux é iniciado, a política é carregada na memória. A política SELinux vem em formato modular, muito parecido com os módulos do kernel carregados no momento da inicialização. E, assim como os módulos do kernel, eles podem ser adicionados e removidos dinamicamente da memória em tempo de execução. O armazenamento de políticas usado pelo SELinux rastreia os módulos que foram carregados. O comando `sestatus` mostra o nome do armazenamento de política. O comando `semodule -l` lista os módulos de política SELinux carregados atualmente na memória.

Para ter uma ideia disso, vamos executar o comando `semodule`:

```
[root@localhost 4linux]# semodule -l | less
```

O resultado será mais ou menos assim:

```
abrt    1.2.0
accountsd       1.0.6
acct    1.5.1
afs     1.8.2
aiccu   1.0.2
aide    1.6.1
ajaxterm        1.0.0
alsa    1.11.4
amanda  1.14.2
amtu    1.2.3
anaconda        1.6.1
antivirus       1.0.0
apache  2.4.0
...
...
```

`semodule` pode ser usado para uma série de outras tarefas, como instalação, remoção, recarregamento, atualização, habilitação e desabilitação dos módulos de política SELinux.

A esta altura, você provavelmente estaria interessado em saber onde os arquivos do módulo estão localizados. A maioria das distribuições modernas inclui versões binárias dos módulos como parte dos pacotes SELinux. Os arquivos de política têm uma extensão .pp. Para CentOS 8, podemos executar o seguinte comando:

```
[root@localhost 4linux]# ls -l /etc/selinux/targeted/modules/active/modules/
```

A lista mostra vários arquivos com a extensão `.pp`. Se você olhar de perto, eles se relacionarão a diferentes aplicativos:

```
...
-rw-r--r--. 1 root root 10692 Aug 20 11:41 anaconda.pp
-rw-r--r--. 1 root root 11680 Aug 20 11:41 antivirus.pp
-rw-r--r--. 1 root root 24190 Aug 20 11:41 apache.pp
-rw-r--r--. 1 root root 11043 Aug 20 11:41 apcupsd.pp
...
```

Os arquivos `.pp` não podem ser lidos por humanos.

A forma como a modularização do SELinux funciona é que, quando o sistema é inicializado, os módulos de política são combinados no que é conhecido como política ativa . Esta política é então carregada na memória. A versão binária combinada desta política carregada pode ser encontrada no diretório `/etc/selinux/targeted/policy`.

```
[root@localhost 4linux]# ls -l /etc/selinux/targeted/policy/
```

irá mostrar a política ativa.

```
total 3428
-rw-r--r--. 1 root root 3510001 Aug 20 11:41 policy.29
```

Alterando as configurações booleanas do SELinux

Embora você não possa ler os arquivos do módulo de política, há uma maneira simples de ajustar suas configurações. Isso é feito por meio de booleanos do SELinux.

Para ver como funciona, vamos executar o comando `semanage boolean -l`.

```
[root@localhost 4linux]# semanage boolean -l | less
```

Isso mostra os diferentes interruptores que podem ser ligados ou desligados, o que eles fazem e seus status atuais:

```
ftp_home_dir                   (off  ,  off)  Allow ftp to home dir
smartmon_3ware                 (off  ,  off)  Allow smartmon to 3ware
mpd_enable_homedirs            (off  ,  off)  Allow mpd to enable homedirs
xdm_sysadm_login               (off  ,  off)  Allow xdm to sysadm login
xen_use_nfs                    (off  ,  off)  Allow xen to use nfs
```

```
mozilla_read_content            (off  ,  off)  Allow mozilla to read content
ssh_chroot_rw_homedirs          (off  ,  off)  Allow ssh to chroot rw homedirs
mount_anyfile                   (on   ,   on)  Allow mount to anyfile
...
...
```

Podemos ver que a primeira opção permite que o daemon do FTP acesse os diretórios pessoais dos usuários. A configuração está desativada no momento.

Para alterar qualquer uma das configurações, podemos usar o comando `setsebool`. Como exemplo, vamos considerar o acesso de gravação de FTP anônimo:

```
[root@localhost 4linux]# getsebool ftpd_anon_write
```

Isso nos mostra que a chave está desligada no momento:

```
[root@localhost 4linux]# ftpd_anon_write --> off
```

Em seguida, alteramos o booleano para habilitá-lo:

```
[root@localhost 4linux]# setsebool ftpd_anon_write on
```

Verificar o valor novamente deve mostrar a mudança:

```
[root@localhost 4linux]# ftpd_anon_write --> on
```

Os booleanos alterados não são permanentes. Eles voltam aos valores antigos após uma reinicialização. Para tornar as coisas permanentes, podemos usar a opção `-P` com o comando `setsebool`.# Diagnosticar e resolver violações de rotina da política do SELinux ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar segurança**
  - Defina as configurações do firewall usando firewall-cmd/firewalld
  - Crie e use listas de controle de acesso a arquivos
  - Configure a autenticação baseada em chave para SSH

- Defina modos de aplicação e permissivos para SELinux
- Liste e identifique o contexto de processos e arquivos do SELinux
- Restaure contextos de arquivo padrão
- Use as configurações booleanas para modificar as configurações do SELinux do sistema
- **Diagnostique e resolva violações de rotina da política do SELinux**

Usuários SELinux

Os usuários do SELinux são entidades diferentes das contas de usuários normais do Linux, incluindo a conta root. Um usuário SELinux não é algo que você cria com um comando especial, nem tem seu próprio acesso de login ao servidor. Em vez disso, os usuários do SELinux são definidos na política que é carregada na memória no momento da inicialização e existem apenas alguns desses usuários. Os nomes de usuário terminam com `_u`, assim como os tipos ou nomes de domínio terminam com `_t` e as funções terminam com `_r`. Diferentes usuários do SELinux têm direitos diferentes no sistema e é isso que os torna úteis.

O usuário SELinux listado na primeira parte do contexto de segurança de um arquivo é o usuário que possui esse arquivo. Um rótulo de usuário em um contexto de processo mostra o privilégio do usuário SELinux com o qual o processo está sendo executado.

Quando o SELinux é imposto, cada conta de usuário regular do Linux é mapeada para uma conta de usuário SELinux. Pode haver várias contas de usuário mapeadas para o mesmo usuário SELinux. Esse mapeamento permite que uma conta regular herde a permissão de sua contraparte SELinux.

Para visualizar esse mapeamento, podemos executar o comando `semanage login -l`:

```
[root@localhost 4linux]# semanage login -l
```

No RHEL, isso é o que podemos ver:

```
Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
system_u             system_u             s0-s0:c0.c1023       *
```

A primeira coluna nesta tabela, "Nome de Login", representa as contas de usuário Linux locais. Mas existem apenas três listados aqui... você pode se perguntar: não criamos

algumas contas durante essas aulas? Sim, e eles são representados pela entrada mostrada como padrão. Qualquer conta de usuário regular do Linux é primeiro mapeada para o login padrão. Isso é então mapeado para o usuário SELinux chamado `unconfined_u`. No nosso caso, esta é a segunda coluna da primeira linha. A terceira coluna mostra a classe de segurança multinível/Segurança de várias categorias (MLS/MCS) para o usuário. Por enquanto, vamos ignorar essa parte e também a coluna depois dela (Serviço).

Em seguida, temos o usuário root. Observe que ele não está mapeado para o login `__default__`, em vez disso, ele recebeu sua própria entrada. Mais uma vez, o root também é mapeado para o usuário SELinux `unconfined_u`.

`system_u` é uma classe diferente de usuário, destinada a executar processos ou daemons.

Para ver quais usuários SELinux estão disponíveis no sistema, podemos executar o comando `semanage user`:

```
[root@localhost 4linux]# semanage user -l
```

A saída em nosso sistema RHEL deve ser semelhante a esta:

```
                Labeling   MLS/       MLS/
SELinux User    Prefix     MCS Level  MCS Range                    SELinux Roles

guest_u         user       s0         s0                           guest_r
root            user       s0         s0-s0:c0.c1023               staff_r sysadm_r
     system_r unconfined_r
staff_u         user       s0         s0-s0:c0.c1023               staff_r sysadm_r
     system_r unconfined_r
sysadm_u        user       s0         s0-s0:c0.c1023               sysadm_r
system_u        user       s0         s0-s0:c0.c1023               system_r unconfined_r
unconfined_u    user       s0         s0-s0:c0.c1023               system_r unconfined_r
user_u          user       s0         s0                           user_r
xguest_u        user       s0         s0                           xguest_r
```

Em primeiro lugar, ele mostra os diferentes usuários do SELinux definidos pela política. Já tínhamos visto usuários como `unconfined_u` e `system_u` antes, mas agora estamos vendo outros tipos de usuários como `guest_u`, `staff_u`, `sysadm_u`, `user_u` e assim por diante. Os nomes são um pouco indicativos dos direitos associados a eles. Por exemplo, podemos assumir que o usuário `sysadm_u` teria mais direitos de acesso do que `guest_u`.

Para verificar nosso convidado, vamos dar uma olhada na quinta coluna, Funções do SELinux. Se você se lembra das primeiras desse assunto, as funções do SELinux são como gateways entre um usuário e um processo. Também os comparamos a filtros: um usuário pode inserir

uma função, desde que a função conceda. Se uma função estiver autorizada a acessar um domínio de processo, os usuários associados a essa função poderão entrar nesse domínio de processo.

Agora, a partir desta tabela, podemos ver que o usuário `unconfined_u` está mapeado para as funções `system_r` e `unconfined_r`. Embora não seja evidente aqui, a política SELinux na verdade permite que essas funções executem processos no domínio `unconfined_t`. Da mesma forma, o usuário `sysadm_u` está autorizado para a função `sysadm r`, mas o guest u é mapeado para a função `guest_r`. Cada uma dessas funções terá diferentes domínios autorizados para elas.

Agora, se dermos um passo para trás, também vimos no primeiro fragmento de código que o login padrão mapeia para o não confinado `usuário u` , assim como o usuário root mapeia para o usuário `unconfined_u`. Uma vez que o login `**default_**` representa qualquer conta de usuário regular do Linux, essas contas serão autorizadas para as funções `system_r` e `unconfined_r` também.

Então, o que isso realmente significa é que qualquer usuário Linux que mapeie para o usuário `unconfined_u` terá os privilégios para executar qualquer aplicativo executado dentro do domínio `unconfined_t`.

Para demonstrar isso, vamos executar o comando `id -Z` como usuário root:

```
[root@localhost 4linux]# id -Z
```

Isso mostra o contexto de segurança SELinux para root :

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Portanto, a conta root é mapeada para o usuário `UNFINUX_U` do SELinux, e `UNOFINED_U` é autorizado para a função `UNFINFIND_R`, que por sua vez está autorizado a executar processos no domínio `UNFINED_T`.

Sugiro que você crie e reserve um tempo agora para iniciar quatro novas sessões SSH com os usuários abaixo em janelas de terminal separadas. Isso nos ajudará a alternar entre contas diferentes quando necessário.

- usuario_regular
- switcheduser
- usuario_convidado

- usuario_restrito

Em seguida, mudamos para a sessão de terminal conectada como usuario_regular. Se você se lembra, criamos várias contas de usuário em aulas passadas, e usuario_regular foi uma delas. Se ainda não o fez, abra uma janela de terminal separada para se conectar ao seu sistema CentOS como usuario_regular. Se executarmos o mesmo comando `id -Z` a partir daí, a saída será semelhante a esta:

```
[usuario_regular@localhost ~]$ id -Z
```

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Nesse caso, a conta regulauser é mapeada para a conta do usuário SELinux unconfined_u e pode assumir a função unconfined_r. A função pode executar processos em um domínio não confinado. Este é o mesmo usuário/função/ domínio SELinux para o qual a conta root também mapeia. Isso porque a política direcionada do SELinux permite que usuários logados executem em domínios não confinados.

Tínhamos visto a lista de vários usuários do SELinux antes:

- **guest_u** : este usuário não tem acesso ao sistema X-Window (GUI) ou rede e não pode executar o comando su / sudo.
- **xguest_u** : este usuário tem acesso às ferramentas GUI e a rede está disponível através do navegador Firefox.
- **user_u** : este usuário tem mais acesso do que as contas de convidado (GUI e rede), mas não pode alternar entre usuários executando su ou sudo.
- **staff_u** : mesmos direitos que user_u, exceto que pode executar o comando sudo para ter privilégios de root.
- **system_u** : este usuário destina-se a executar serviços do sistema e não deve ser mapeado para contas de usuário regulares.

SELinux em ação 1: restringindo o acesso de usuário comutado

Para ver como o SELinux pode reforçar a segurança para contas de usuário, vamos pensar sobre a conta de usuario_regular. Como administrador do sistema, você agora sabe que o usuário tem os mesmos privilégios SELinux irrestritos da conta root e gostaria de mudar isso. Especificamente, você não deseja que o usuário seja capaz de alternar para outras contas, incluindo a conta root.

Vamos primeiro verificar a capacidade do usuário de alternar para outra conta. No fragmento de código a seguir, o usuario_regular muda para a conta do switcheduser. Presumimos que ele conheça a senha de switcheduser:

```
[usuario_regular@localhost ~]$ su - switcheduser
Password:
[switcheduser@localhost ~]$
```

Em seguida, voltamos para a janela do terminal conectado como usuário root e alteramos o mapeamento do usuário SELinux do usuario_regular. Mapearemos usuario_regular para `user_u`.

```
semanage login -a -s user_u usuario_regular
```

Então, o que estamos fazendo aqui? Estamos adicionando (-a) a conta do usuario_regular à conta de usuário do SELinux (-s) `user_u`. A mudança não terá efeito até que o usuario_regular efetue logout e login novamente.

Voltando à janela do terminal do usuario_regular, primeiro mudamos de switcheduser:

```
[switcheduser@localhost ~]$ logout
```

Em seguida, o usuario_regular também se desconecta:

```
[usuario_regular@localhost ~]$ logout
```

Abrimos então uma nova janela de terminal para conectar como usuario_regular. Depois, tentamos mudar para switchuser novamente:

```
[usuario_regular@localhost ~]$ su - switcheduser
Password:
```

Isso é o que vemos agora:

```
su: Authentication failure
```

Se executarmos o comando `id -Z` novamente para ver o contexto SELinux para usuario_regular, veremos que a saída é bem diferente de antes: usuario_regular agora está mapeado para `user_u`.

```
[usuario_regular@localhost ~]$ id -Z
```

```
user_u:user_r:user_t:s0
```

Então, onde você usaria essas restrições? Você pode pensar em uma equipe de desenvolvimento de aplicativos dentro de sua organização de TI. Você pode ter vários desenvolvedores e testadores nessa equipe, codificando e testando o aplicativo mais recente para sua empresa. Como administrador de sistema, você sabe que os desenvolvedores estão mudando de suas contas para algumas das contas de alto privilégio para fazer alterações ad-hoc em seu servidor. Você pode impedir que isso aconteça restringindo sua capacidade de trocar de conta. (Mas lembre-se, isso ainda não os impede de fazer login diretamente como o usuário com altos privilégios).

SELinux em ação 2: restringindo permissões para executar scripts

Vejamos outro exemplo de restrição de acesso do usuário por meio do SELinux. Execute esses comandos a partir da sessão raiz.

Por padrão, o SELinux permite que os usuários mapeados para a conta `guest_t` executem scripts a partir de seus diretórios pessoais. Podemos executar o comando `getsebool` para verificar o valor booleano:

```
getsebool allow_guest_exec_content
```

A saída mostra que o sinalizador está ativo.

```
guest_exec_content --> on
```

Para verificar seu efeito, vamos primeiro alterar o mapeamento do usuário SELinux para a

conta usuario_convidado que criamos no início deste tutorial. Faremos isso como usuário root.

```
semanage login -a -s guest_u usuario_convidado
```

Podemos verificar a ação executando o comando `semanage login -l` novamente:

```
semanage login -l
```

Como podemos ver, usuario_convidado agora está mapeado para a conta de usuário guest_u SELinux.

```
Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
usuario_convidado            guest_u              s0                    *
usuario_regular          user_u               s0                  *
root                 unconfined_u         s0-s0:c0.c1023       *
system_u             system_u             s0-s0:c0.c1023       *
```

Se tivermos uma janela de terminal aberta como usuario_convidado, sairemos dela e faremos o logon novamente em uma nova janela de terminal como usuario_convidado.

A seguir, criaremos um script bash extremamente simples no diretório inicial do usuário. Os blocos de código a seguir verificam primeiro o diretório inicial, depois criam o arquivo e o lêem no console. Finalmente, a permissão de execução é alterada.

Verifique se você está no diretório `usuario_convidado` inicial:

```
[usuario_convidado@localhost ~]$ pwd
/home/usuario_convidado
```

Crie o script:

```
[usuario_convidado@localhost ~]$ vi myscript.sh
```

Conteúdo do script:

```
#!/bin/bash
echo "This is a test script"
```

Torne o script executável:

```
chmod u+x myscript.sh
```

Quando tentamos executar o script como usuario_convidado, ele funciona conforme o esperado:

```
[usuario_convidado@localhost ~]$ ~/myscript.sh
This is a test script
```

Em seguida, voltamos para a janela do terminal raiz e alteramos a configuração booleana allow_guest_exec_content para `off` e verificamos:

```
setsebool allow_guest_exec_content off
getsebool allow_guest_exec_content
guest\_exec\_content --> off
```

Voltando ao console logado como usuario_convidado, tentamos executar o script novamente. Desta vez, o acesso é negado:

```
[usuario_convidado@localhost ~]$ ~/myscript.sh
-bash: /home/usuario_convidado/myscript.sh: Permission denied
```

Portanto, é assim que o SELinux pode aplicar uma camada adicional de segurança sobre o DAC. Mesmo quando o usuário tem acesso total de leitura, gravação e execução ao script criado em seu próprio diretório inicial, ele ainda pode ser impedido de executá-lo. Onde você precisa disso? Bem, pense em um sistema de produção. Você sabe que os desenvolvedores têm acesso a ele, assim como alguns dos contratantes que trabalham para sua empresa. Você gostaria que eles acessassem o servidor para visualizar mensagens de erro e arquivos de log, mas não deseja que executem scripts de shell. Para fazer isso, você pode primeiro habilitar o

SELinux e, em seguida, garantir que o valor booleano correspondente seja definido.

Falaremos sobre as mensagens de erro do SELinux em breve, mas por enquanto, se estivermos ansiosos para ver onde essa negação foi registrada, podemos olhar o arquivo `/var/log/messages`. Execute a partir da sessão raiz:

```
grep "SELinux is preventing" /var/log/messages
```

As duas últimas mensagens no arquivo em nosso servidor CentOS mostram a negação de acesso:

```
Aug 23 12:59:42 localhost setroubleshoot: SELinux is preventing /usr/bin/bash from execute
     access on the file . For complete SELinux messages. run sealert -l 8343a9d2-ca9d-49db
     -9281-3bb03a76b71a
Aug 23 12:59:42 localhost python: SELinux is preventing /usr/bin/bash from execute access
     on the file .
```

A mensagem também mostra um valor de ID longo e sugere que executemos o comando `sealert` com esse ID para obter mais informações. O seguinte comando mostra isso (use seu próprio ID de alerta):

```
sealert -l 8343a9d2-ca9d-49db-9281-3bb03a76b71a
```

E, de fato, a saída nos mostra mais detalhes sobre o erro:

```
SELinux is preventing /usr/bin/bash from execute access on the file .

*****  Plugin catchall_boolean (89.3 confidence) suggests   ******************

If you want to allow guest to exec content
Then you must tell SELinux about this by enabling the 'guest\_exec\_content' boolean.
You can read 'None' man page for more details.
Do
setsebool -P guest\_exec\_content 1

*****  Plugin catchall (11.6 confidence) suggests   *************************

...
```

É uma grande quantidade de saída, mas observe as poucas linhas no início:

O SELinux está impedindo que /usr/bin/bash execute o acesso ao arquivo.

Isso nos dá uma boa ideia de onde o erro está vindo.

As próximas linhas também mostram como corrigir o erro:

```
If you want to allow guest to exec content
Then you must tell SELinux about this by enabling the 'guest\_exec\_content' boolean.
...
setsebool -P guest\_exec\_content 1
```

SELinux em ação 3: restringindo o acesso aos serviços

Já falamos sobre as funções do SELinux quando introduzimos a terminologia básica de usuários, funções, domínios e tipos. Vamos agora ver como as funções também desempenham um papel na restrição do acesso do usuário. Como dissemos antes, uma função no SELinux fica entre o usuário e o domínio do processo e controla em quais domínios o processo do usuário pode entrar. As funções não são tão importantes quando as vemos em contextos de segurança de arquivos. Para arquivos, é listado com um valor genérico de object_r. As funções tornam-se importantes ao lidar com usuários e processos.

Vamos primeiro ter certeza de que o daemon httpd não está rodando no sistema. Como usuário root, você pode executar o seguinte comando para garantir que o processo seja interrompido:

```
service httpd stop
```

Em seguida, mudamos para a janela do terminal em que efetuamos login como usuario_restrito e tentamos ver o contexto de segurança do SELinux para ela. Se você não tiver a janela do terminal aberta, inicie uma nova sessão de terminal no sistema e faça login como a conta de usuário restrita que criamos no início deste tutorial.

```
[usuario_restrito@localhost ~]$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Portanto, a conta tem o comportamento padrão de execução como usuário unconfined_u e tendo acesso à função unconfined_r. No entanto, esta conta não tem o direito de iniciar nenhum processo no sistema. O seguinte bloco de código mostra que o usuario_restrito está tentando iniciar o daemon httpd e obtendo um erro de acesso negado:

```
[usuario_restrito@localhost ~]$ service httpd start
Redirecting to /bin/systemctl start  httpd.service
Failed to issue method call: Access denied
```

Em seguida, voltamos para a janela do terminal do usuário root e nos certificamos de que a conta de usuario_restrito foi adicionada ao arquivo /etc/sudoers. Esta ação permitirá que a conta de usuario_restrito use privilégios de root.

```
visudo
```

E então, no arquivo, adicione a seguinte linha, salve e saia:

```
usuario_restrito ALL=(ALL)      ALL
```

Se sairmos da janela de terminal de usuario_restrito e fizermos login novamente, podemos iniciar e parar o serviço httpd com privilégios sudo:

```
[usuario_restrito@localhost ~]$ sudo service httpd start
```

```
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for usuario_restrito:
Redirecting to /bin/systemctl start  httpd.service
```

O usuário também pode interromper o serviço agora:

```
[usuario_restrito@localhost ~]$ sudo service httpd stop
Redirecting to /bin/systemctl stop  httpd.service
```

Isso tudo é muito normal: os administradores de sistema dão acesso ao sudo para contas de usuário em que eles confiam. Mas e se você quiser impedir que esse usuário específico inicie

o serviço httpd mesmo quando a conta do usuário estiver listada no arquivo sudoers?

Para ver como isso pode ser alcançado, vamos voltar para a janela do terminal do usuário root e mapear o usuario_restrito para a conta user_r SELinux. Isso é o que fizemos para a conta de usuário regular em outro exemplo.

```
semanage login -a -s user_u usuario_restrito
```

Voltando à janela do terminal do usuario_restrito, fazemos logout e entramos novamente em uma nova sessão de terminal como usuario_restrito.

Agora que o usuario_restrito foi restringido a user_u (e isso significa a função user_r e domínio user_t), podemos verificar seu acesso usando o comando `seinfo` de nossa janela de usuário root:

```
seinfo -uuser_u -x
```

A saída mostra as funções que o user_u pode assumir. Estes são object_r e user_r:

```
   user_u
      default level: s0
      range: s0
      roles:
         object_r
         user_r
```

Indo um passo adiante, podemos executar o seinfocomando para verificar em quais domínios a função user_r está autorizada a entrar:

```
seinfo -ruser_r -x
```

Existem vários domínios que o user_r está autorizado a entrar:

```
   user_r
      Dominated Roles:
         user_r
      Types:
         git_session_t
         sandbox_x_client_t
```

```
          git_user_content_t
          virt_content_t
          policykit_grant_t
          httpd_user_htaccess_t
          telepathy_mission_control_home_t
          qmail_inject_t
          gnome_home_t
          ...
          ...
```

Mas esta lista mostra httpd_t como um dos domínios? Vamos tentar o mesmo comando com um filtro:

```
seinfo -ruser_r -x | grep httpd
```

Existem vários domínios relacionados a httpd aos quais a função tem acesso, mas httpd_t não é um deles:

```
          httpd_user_htaccess_t
          httpd_user_script_exec_t
          httpd_user_ra_content_t
          httpd_user_rw_content_t
          httpd_user_script_t
          httpd_user_content_t
```

Tomando este exemplo então, se a conta de usuario_restrito tentar iniciar o daemon httpd, o acesso deve ser negado porque o processo httpd é executado dentro do domínio httpd_t e este não é um dos domínios que a função user_r está autorizada a acessar. E sabemos que user_u (mapeado para strictuser) pode assumir a função user_r. Isso deve falhar mesmo se a conta de usuario_restrito tiver o privilégio sudo concedido.

Voltando à janela do terminal da conta de usuario_restrito, tentamos iniciar o daemon httpd agora (fomos capazes de interrompê-lo antes porque a conta recebeu o privilégio sudo):

```
[usuario_restrito@localhost ~]$ sudo service httpd start
```

O acesso é negado:

`shell sudo: PERM_SUDOERS: setresuid(-1, 1, -1): Operation not permitted`# Localizar e recuperar imagens de container em um registro remoto ## Pontos de estudo para o exame Os candidatos

ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar contêineres**
  - **Encontre e recupere imagens de contêiner de um registro remoto**
  - Inspecione imagens de contêineres
  - Execute o gerenciamento de contêineres usando comandos como podman e skopeo
  - Execute o gerenciamento básico de contêineres, como executar, iniciar, interromper e listar contêineres em execução
  - Execute um serviço dentro de um contêiner
  - Configure um contêiner para iniciar automaticamente como um serviço systemd
  - Anexe armazenamento persistente a um contêiner

Introdução

Um contêiner Linux é um conjunto de 1 ou mais processos isolados do resto do sistema. Todos os arquivos necessários para executá-los são fornecidos a partir de uma imagem distinta, o que significa que os contêineres do Linux são portáteis e consistentes à medida que passam do desenvolvimento para o teste e, finalmente, para a produção. Isso os torna muito mais rápidos de usar do que os pipelines de desenvolvimento que dependem da replicação de ambientes de teste tradicionais. Devido à sua popularidade e facilidade de uso, os contêineres também são uma parte importante da segurança de TI.

Por que usar contêineres Linux?

Imagine que você está desenvolvendo em algum aplicativo, ou configuração de algum serviço. Você realiza seu trabalho em um notebook e seu ambiente possui uma configuração específica. No entanto, outras pessoas da sua equipe podem ter configurações um tanto quanto diferentes. O aplicativo ou o serviço que você está trabalhando depende dessa configuração, das suas bibliotecas, dependências e arquivos específicos. Enquanto isso, sua empresa possui ambientes de desenvolvimento e produção padronizados com suas próprias configurações e seus próprios conjuntos de arquivos de suporte. Você deseja emular esses ambientes o máximo possível localmente, mas sem toda a sobrecarga de recriar os ambientes de servidor. Então, como você faz seu aplicativo funcionar nesses ambientes, passar pela garantia de qualidade e implementar seu aplicativo sem grandes dores de cabeça, reescrita e conserto de falhas? A resposta: contêineres.

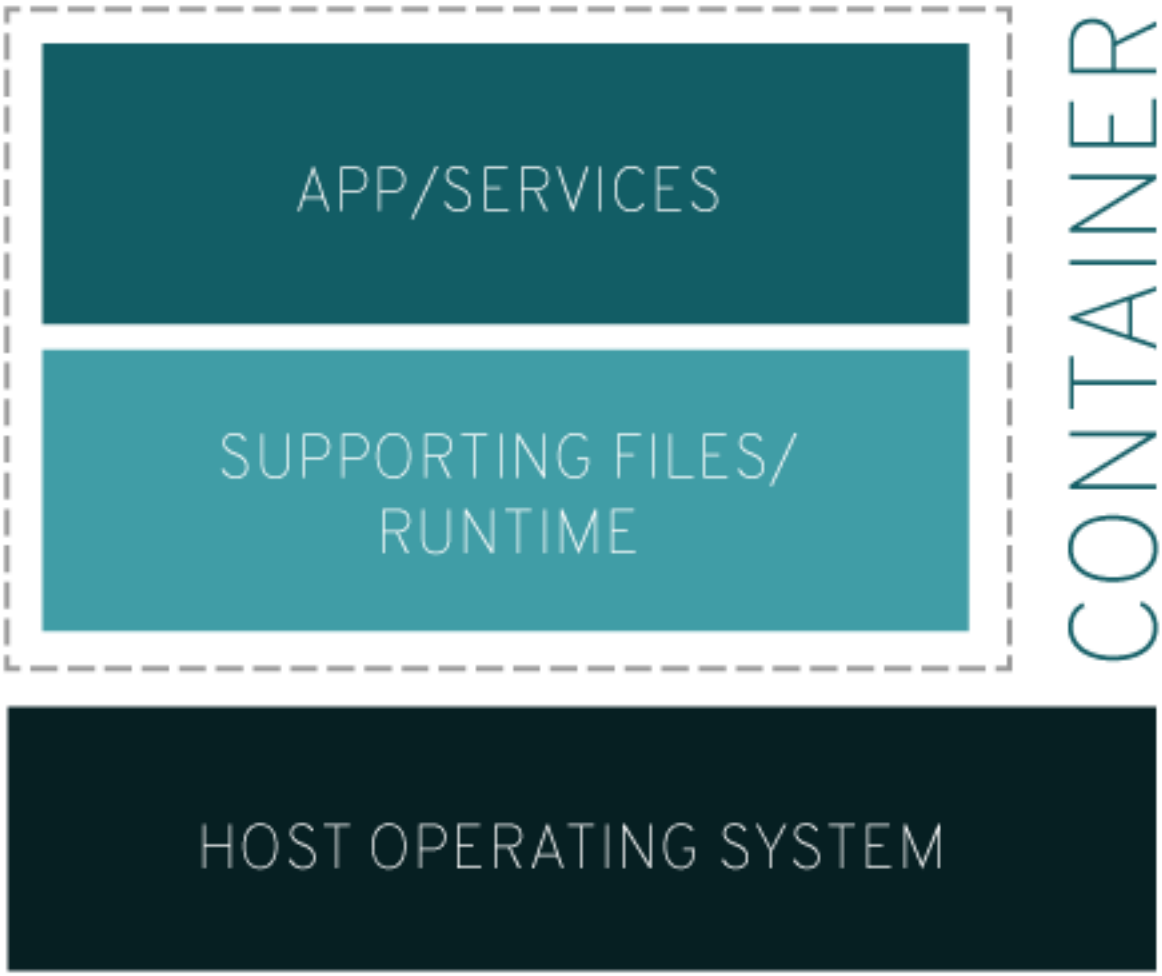


Fig. 28.1: Containers

O contêiner que contém seu aplicativo tem as bibliotecas, dependências e arquivos necessários para que você possa movê-lo durante a produção sem efeitos colaterais desagradáveis. Na verdade, o conteúdo de uma imagem de contêiner pode ser considerado uma instalação de uma distribuição Linux porque vem completo com pacotes RPM, arquivos de configuração, etc. Mas a distribuição de imagens de contêiner é muito mais fácil do que instalar novas cópias de sistemas operacionais.

Esse é um exemplo comum, mas os contêineres Linux podem ser aplicados a muitos problemas diferentes onde portabilidade, configurabilidade e isolamento são necessários. O objetivo dos contêineres Linux é desenvolver mais rapidamente e atender às necessidades de negócios conforme elas surgem. Em alguns casos, os contêineres são essenciais porque são a única maneira de fornecer a escalabilidade de que um aplicativo precisa. Não importa a infraestrutura - local, na nuvem ou um híbrido das duas - os contêineres atendem à demanda. Obviamente, escolher a plataforma de contêiner certa é tão importante quanto os próprios contêineres.

Começando com contêineres

Os contêineres do Linux surgiram como um pacote de aplicativos de código aberto chave e tecnologia de entrega, combinando o isolamento de aplicativos leves com a flexibilidade dos métodos de implantação baseados em imagem. `RHEL` implementa contêineres Linux usando tecnologias básicas, como:

- Grupos de controle (cgroups) para gerenciamento de recursos
- Namespaces para isolamento de processos
- `SELinux` para segurança
- Multilocação segura

Essas tecnologias reduzem o potencial de explorações de segurança e fornecem um ambiente para a produção e execução de contêineres de qualidade empresarial.

O Red Hat OpenShift fornece ferramentas poderosas de linha de comando e IU da Web para criar, gerenciar e executar contêineres em unidades chamadas de `pods`. O Red Hat permite que você construa e gerencie containers individuais e imagens de container fora do OpenShift. A Red Hat tambem fornece um conjunto de ferramentas de linha de comando que podem operar sem um mecanismo de contêiner. Esses incluem: Um mecanismo de contêiner sem daemon que gerencia diretamente contêineres e imagens de contêiner (`run`,`stop`,`start`,`ps`,`attach` ,`exec`, e assim por diante). - **buildah** - É usado para criar novas imagens de contêiner. É uma ferramenta que facilita a construção de imagens de contêineres OCI. - **skopeo** - É usado para inspecionar, copiar, excluir e assinar imagens de contêiner. - **runc** - Para fornecer recursos de execução e construção de contêiner para `podman` e `buildah` - **crun** - Um tempo de execução opcional que pode ser configurado e oferece maior flexibilidade, controle e segurança para contêineres sem raiz.

Como essas ferramentas são compatíveis com a Open Container Initiative (OCI), elas podem ser usadas para gerenciar os mesmos contêineres Linux que são produzidos e gerenciados pelo Docker e outros mecanismos de contêiner compatíveis com OCI . No entanto, eles são especialmente adequados para rodar diretamente no Red Hat Enterprise Linux, em casos de uso de nó único.

Executar contêineres sem Docker

A Red Hat removeu o mecanismo de contêiner `Docker` e o comando docker do **RHEL 8**.

As ferramentas `Podman`, `Skopeo` e `Buildah` foram desenvolvidas para substituir os recursos de comando do `Docker`. Cada ferramenta neste cenário é mais leve e focada em um subconjunto de recursos.

As principais vantagens das ferramentas `Podman`, `Skopeo` e `Buildah` incluem:

- Executando no modo sem root - Os contêineres sem root são muito mais seguros, pois são executados sem quaisquer privilégios adicionais.
- Nenhum daemon necessário - Essas ferramentas têm requisitos de recursos muito mais baixos em inatividade, pois quando você não está executando contêineres, o `Podman` não está em execução em vez de ter um daemon sempre em execução
- Integração nativa do systemd - O `Podman` permite que você crie arquivos de unidade do `systemd` e execute contêineres como serviços do sistema

Se ainda sim não foi convencido e quer usar o `Docker` no **RHEL**, você pode obtê-lo de diferentes projetos upstream, mas não é compatível com o RHEL 8.

Registro de contêineres

Este é um repositório para armazenar e recuperar imagens de contêiner. As imagens podem ser carregadas ou enviadas para um registro de contêiner. Esses contêineres podem ser obtidos ou baixados do registro para um sistema local para que você possa usá-los para executar contêineres. Existem dois tipos de registros: privado e público. Um registro público contém principalmente imagens de terceiros, enquanto os registros privados são controlados por uma organização.

A Red Hat distribui imagens de contêiner certificadas por meio de dois registros principais acessados por meio das credenciais de login da Red Hat anexadas à sua assinatura:

- **registry.connect.redhat.com** - Para contêineres baseados em produtos de terceiros.
- **registry.redhat.io** - Para containers baseados em produtos oficiais da Red Hat.

Convenções de nomenclatura para imagens de contêiner

As imagens seguem a seguinte sintaxe de nomenclatura FQDN:

```
registry_name/user_name/image_name:tag
```

Vamos analisar a sintaxe do FQDN:

- **registry_name** - Nome do registro que armazena a imagem, ou seja, o FQDN do registro.
- **user_name** - Representa a organização à qual a imagem pertence.
- **image_name** - Nome da imagem e deve ser exclusivo no namespace.
- **tag** - Usada para identificar a versão da imagem.

Hands On

- Instalar Podman/Container-tools no RHEL8

```
[root@rhel8 joatham]# yum module install container-tools
Updating Subscription Management repositories.
Última verificação de data de vencimento de metadados: 0:01:52 atrás em ter 12 out 2021
     20:54:23 -03.
Dependências resolvidas.
================================================================================================================

 Pacote    Arq.     Versão                                        Repositório
                             Tamanho
================================================================================================================

Instalando grupo/pacotes do módulo:
 crun      x86_64   0.20.1-1.module+el8.4.0+11822+6cc1e7d7     rhel-8-for-x86_64-appstream-
      rpms   192 k
 skopeo    x86_64   1:1.3.1-5.module+el8.4.0+11990+22932769    rhel-8-for-x86_64-appstream-
      rpms   7.0 M
 toolbox   noarch   0.0.8-1.module+el8.4.0+11822+6cc1e7d7      rhel-8-for-x86_64-appstream-
      rpms    16 k
 udica     noarch   0.2.4-2.module+el8.4.0+11822+6cc1e7d7      rhel-8-for-x86_64-appstream-
      rpms    51 k
Instalando perfis de módulo:
 container-tools/common


Resumo da transação
================================================================================================================

Instalar  4 Pacotes

Tamanho total do download: 7.3 M
Tamanho depois de instalado: 26 M
Correto? [s/N]: s
```

```
[root@rhel8 containers]# podman version
Version:      3.2.3
API Version:  3.2.3
Go Version:   go1.15.7
Built:        Thu Jul 29 12:02:43 2021
OS/Arch:      linux/amd64
```

```
[root@rhel8 containers]# podman --help
Manage pods, containers and images

Usage:
  podman [options] [command]

Available Commands:
  attach      Attach to a running container
  auto-update Auto update containers according to their auto-update policy
  build       Build an image using instructions from Containerfiles
```

```
  commit      Create new image based on the changed container
  container   Manage containers
  cp          Copy files/folders between a container and the local filesystem
  create      Create but do not start a container
  diff        Display the changes to the object's file system
  events      Show podman events
  exec        Run a process in a running container
  export      Export container's filesystem contents as a tar archive
  generate    Generate structured data based on containers, pods or volumes.
  healthcheck Manage health checks on containers
  help        Help about any command
  history     Show history of a specified image
  image       Manage images
  images      List images in local storage
  import      Import a tarball to create a filesystem image
  info        Display podman system information
  init        Initialize one or more containers
  inspect     Display the configuration of object denoted by ID
  kill        Kill one or more running containers with a specific signal
  load        Load image(s) from a tar archive
  logout      Logout of a container registry
  logs        Fetch the logs of one or more containers
  machine     Manage a virtual machine
  manifest    Manipulate manifest lists and image indexes
  mount       Mount a working container's root filesystem
  network     Manage networks
  pause       Pause all the processes in one or more containers
  play        Play containers, pods or volumes from a structured file.
```

- Conferir repositórios dentro de `repositories.conf`

```
[root@rhel8 containers]# vi /etc/containers/registries.conf
...
# To ensure compatibility with docker we've included docker.io in the default search list.
     However Red Hat
# does not curate, patch or maintain container images from the docker.io registry.
[registries.search]
registries = ['registry.access.redhat.com', 'registry.redhat.io', 'docker.io']
unqualified-search-registries = ["registry.fedoraproject.org", "registry.access.redhat.com
     ", "registry.centos.org", "docker.io"]
...
```

- Conferir se o repositório está sendo acionado:

```
[root@rhel8 containers]# podman search httpd | less

INDEX       NAME
     DESCRIPTION                                     STARS       OFFICIAL    AUTOMATED
redhat.com  registry.access.redhat.com/rhscl/httpd-24-rhel7
     Apache HTTP 2.4 Server                          0
redhat.com  registry.access.redhat.com/cloudforms46-beta/cfme-openshift-httpd
     CloudForms is a management and automation pl...  0
redhat.com  registry.access.redhat.com/cloudforms46/cfme-openshift-httpd
```

```
     Web Server image for a multi-pod Red Hat® C...   0
redhat.com  registry.access.redhat.com/rhmap43/httpd
     Provides an extension to the RHSCL Httpd Doc...  0
redhat.com  registry.access.redhat.com/rhmap47/httpd
     Provides an extension to the RHSCL Httpd ima...  0
redhat.com  registry.access.redhat.com/ubi8/httpd-24
     Platform for running Apache httpd 2.4 or bui...  0
```

- Instalando/Consultando/Removendo uma imagem do repositorio remoto RedHat

```
[root@rhel8 storage]# podman pull registry.access.redhat.com/rhmap42/httpd
Trying to pull registry.access.redhat.com/rhmap42/httpd:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob d77ab047cab8 done
Copying blob a720563b6d37 done
Copying blob f9ab7164e76a done
Copying blob 5ef19e961bd1 done
Copying blob aad5d4c7bba9 done
Writing manifest to image destination
Storing signatures
5fae29391474293e20a7bfe3e4f4d0d52fd60a787119f4b8946c50cd9e073623
```

```
[root@rhel8 storage]# podman images
REPOSITORY                                TAG         IMAGE ID      CREATED      SIZE
registry.access.redhat.com/rhmap42/httpd  latest      5fae29391474  4 years ago  480 MB
```

```
[root@rhel8 storage]# podman rmi -a
Untagged: registry.access.redhat.com/rhmap42/httpd:latest
Deleted: 5fae29391474293e20a7bfe3e4f4d0d52fd60a787119f4b8946c50cd9e073623
[root@rhel8 storage]# podman images
REPOSITORY  TAG         IMAGE ID    CREATED     SIZE
```

29

Inspecionar imagens de container

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar contêineres**
  - Encontre e recupere imagens de contêiner de um registro remoto
  - **Inspecione imagens de contêineres**
  - Execute o gerenciamento de contêineres usando comandos como podman e skopeo
  - Execute o gerenciamento básico de contêineres, como executar, iniciar, interromper e listar contêineres em execução
  - Execute um serviço dentro de um contêiner
  - Configure um contêiner para iniciar automaticamente como um serviço systemd
  - Anexe armazenamento persistente a um contêiner

O que seria uma imagem de um contêiner

A imagem do contêiner é um binário que inclui todos os requisitos para executar um único contêiner e metadados que descrevem suas necessidades e recursos.

Existem dois tipos:

- Imagens de base do Red Hat Enterprise Linux (imagens de base do RHEL)

- Imagens de base universal da Red Hat (imagens UBI)

Ambos os tipos são construídos a partir de partes do Sistema Operacional RHEL. Nesse caso, os usuários podem se beneficiar de grande confiabilidade, segurança, desempenho e ciclos de vida.

A principal diferença entre esses dois tipos é que as imagens UBI permitem que você compartilhe imagens de contêiner com outras pessoas. Você pode construir um aplicativo em contêiner usando UBI, colocá-lo no servidor de registro de sua escolha, compartilhá-lo facilmente com outras pessoas e até mesmo implantá-lo em plataformas que não facam parte do Red Hat. As imagens UBI são projetadas para ser uma base para casos de uso de aplicativos da web e nativos da nuvem desenvolvidos em contêineres.

Características das imagens RHEL

As seguintes características se aplicam a imagens de base RHEL e imagens UBI.

Em geral, as imagens de contêiner RHEL são:

- **Supported**: suportado pela Red Hat para uso com aplicativos em contêineres. Eles contêm os mesmos pacotes de software protegidos, testados e certificados encontrados no Red Hat Enterprise Linux.
- **Cataloged**: listado no Red Hat Container Catalog, com descrições, detalhes técnicos e um índice de integridade para cada imagem.
- **Updated**: oferecido com um cronograma de atualização bem definido, para obter o software mais recente.
- **Tracked**: rastreado pela Errata do Produto Red Hat para ajudar a entender as mudanças que são adicionadas a cada atualização.
- **Reusable**: as imagens do contêiner precisam ser baixadas e armazenadas em cache em seu ambiente de produção uma vez. Cada imagem de contêiner pode ser reutilizada por todos os contêineres que a incluem como base.

Características das imagens UBI

As imagens UBI permitem que você compartilhe imagens de contêiner com outras pessoas. Quatro imagens UBI são oferecidas: `micro`, `minimal`, `standard` e `init`.

As seguintes características se aplicam às imagens UBI:

- **Construído a partir de um subconjunto de conteúdo RHEL**: as imagens do Red Hat Universal Base são construídas a partir de um subconjunto de conteúdo normal do Red Hat Enterprise Linux.

- **Redistribuível**: as imagens UBI permitem a padronização para clientes, parceiros, ISVs e outros da Red Hat. Com as imagens UBI, você pode construir suas imagens de contêiner em uma base de software oficial Red Hat que pode ser livremente compartilhado e implantado.
- **Fornece um conjunto de quatro imagens de base**: micro, minimal, standard e init.
- **Fornece um conjunto de imagens de contêineres de tempo de execução de linguagem pré-construídas**: as imagens de tempo de execução baseadas em Application Streams fornecem uma base para aplicativos que podem se beneficiar de tempos de execução padrão suportados, como python, perl, php, dotnet, nodejs e ruby.
- **Fornece um conjunto de repositórios YUM associados**: os repositórios YUM incluem pacotes RPM e atualizações que permitem adicionar dependências de aplicativos e reconstruir imagens de contêiner UBI.
  - O repositório ubi-8-baseos contém o subconjunto redistribuível de pacotes RHEL que você pode incluir em seu contêiner.
  - O repositório ubi-8-appstream contém pacotes de fluxos de aplicativos que você pode adicionar a uma imagem UBI para ajudá-lo a padronizar os ambientes que você usa com aplicativos que requerem tempos de execução específicos.
  - **Adicionando RPMs UBI**: Você pode adicionar pacotes RPM a imagens UBI de repositórios UBI pré-configurados.
- **Licenciamento**: Você é livre para usar e redistribuir imagens UBI, desde que cumpra o Contrato de Licença de Usuário Final do Red Hat Universal Base Image.

Inspeção de imagens locais - podman

Depois de "baixar" uma imagem para seu sistema local e antes de executá-la, uma boa prática é dar uma investigada nessa imagem. As razões para investigar incluem:

- Entender o que a imagem faz
- Verificar que software está dentro da imagem
- Procedimento

O comando podman inspect exibe informações básicas sobre o que uma imagem faz.

```
[root@rhel8 storage]# podman pull registry.access.redhat.com/rhmap42/httpd
Trying to pull registry.access.redhat.com/rhmap42/httpd:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob a720563b6d37 done
Copying blob d77ab047cab8 done
Copying blob 5ef19e961bd1 done
Copying blob f9ab7164e76a done
Copying blob aad5d4c7bba9 done
Writing manifest to image destination
Storing signatures
```

```
5fae29391474293e20a7bfe3e4f4d0d52fd60a787119f4b8946c50cd9e073623
```

```
[root@rhel8 storage]# podman inspect registry.access.redhat.com/rhmap42/httpd | less

[
    {
        "Id": "5fae29391474293e20a7bfe3e4f4d0d52fd60a787119f4b8946c50cd9e073623",
        "Digest": "sha256:a057bc64c8373538065162cc08eef9d63161ae9158744380eecabd1ca75ef12e
            ",
        "RepoTags": [
            "registry.access.redhat.com/rhmap42/httpd:latest"
        ],
        "RepoDigests": [
            "registry.access.redhat.com/rhmap42/httpd@sha256:
                 a057bc64c8373538065162cc08eef9d63161ae91587
44380eecabd1ca75ef12e"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2016-12-13T15:27:41.060362Z",
        "Config": {
            "ExposedPorts": {
                "443/tcp": {},
                "80/tcp": {},
                "8080/tcp": {},
                "8443/tcp": {}
            },
            "Env": [
                "PATH=/opt/app-root/src/bin:/opt/app-root/bin:/usr/local/sbin:/usr/local/
                     bin:/usr/sbin:/
usr/bin:/sbin:/bin",
                "container=docker",
                "STI_SCRIPTS_URL=image:///usr/libexec/s2i",
                "STI_SCRIPTS_PATH=/usr/libexec/s2i",
                "HOME=/opt/app-root/src",
```

Outra opção seria montar a imagem e então inspencioná-la:

```
[root@rhel8 storage]# podman run -d registry.access.redhat.com/rhmap42/httpd
74faf4485ee5d6a24f3d7b6fbce890e888d48c360ccb646f7dbede61f523e726
```

```
[root@rhel8 storage]# podman ps
CONTAINER ID  IMAGE                                           COMMAND
     CREATED         STATUS            PORTS       NAMES
74faf4485ee5  registry.access.redhat.com/rhmap42/httpd:latest  /opt/rh/httpd24/r...  10
     seconds ago  Up 9 seconds ago              keen_hellman
```

```
[root@rhel8 storage]# podman mount 74faf4485ee5
/var/lib/containers/storage/overlay/
```

```
     d55a982cd617a23d1d1b61ebf103f85dc20f43f8424fc0953095b37c166eac23/merged
[root@rhel8 storage]# ls /var/lib/containers/storage/overlay/
     d55a982cd617a23d1d1b61ebf103f85dc20f43f8424fc0953095b37c166eac23/merged
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin
     srv  sys  tmp  usr  var
[root@rhel8 storage]#
```

Ou, ainda, verificar os pacotes listados no container com o comando `rpm`

```
[root@rhel8 storage]# rpm -qa --root=/var/lib/containers/storage/overlay/
     d55a982cd617a23d1d1b61ebf103f85dc20f43f8424fc0953095b37c166eac23/merged
postgresql-devel-9.2.18-1.el7.x86_64
tzdata-2016j-1.el7.noarch
gettext-0.18.2.1-4.el7.x86_64
setup-2.8.71-7.el7.noarch
bzip2-1.0.6-13.el7.x86_64
basesystem-10.0-7.el7.noarch
libcurl-devel-7.29.0-35.el7.x86_64
glibc-common-2.17-157.el7_3.1.x86_64
unzip-6.0-16.el7.x86_64
glibc-2.17-157.el7_3.1.x86_64
make-3.82-23.el7.x86_64
ncurses-libs-5.9-13.20130511.el7.x86_64
wget-1.14-13.el7.x86_64
libsepol-2.5-6.el7.x86_64
hostname-3.13-3.el7.x86_64
```

Inspeção de imagens remotas - skopeo

Segundo seu manual:

```
NAME
       skopeo -- Command line utility used to interact with local and remote container
            images and container image registries
```

Ou seja, para inspecionar a imagem de um container antes de baixá-la para seu sistema, você pode usar o comando `skopeo inspect`. Com ele, você consegue exibir informações sobre uma imagem que reside em um registro remoto de contêineres.

```
shell [root@rhel8 storage]# podman search postgresql | less [root@rhel8 storage]# skopeo inspect
  docker://registry.access.redhat.com/rhscl/postgresql-95-rhel7 ... "Created": "2019-05-31T04:56:10.721167
Z", "DockerVersion": "1.13.1", "Labels": { "architecture": "x86_64", "authoritative-source-url
": "registry.access.redhat.com", "build-date": "2019-05-31T04:54:34.533003", "com.redhat.build
-host": "cpt-0013.osbs.prod.upshift.rdu2.redhat.com", "com.redhat.component": "rh-postgresql95
-container", "com.redhat.license_terms": "https://www.redhat.com/en/about/red-hat-end-user-license
```

-agreements", "description": "PostgreSQL is an advanced Object-Relational database management system (DBMS). The image contains the client and server programs that you'll need to create, run, maintain and access a PostgreSQL DBMS server.", "distribution-scope": "public", "io.k8s.description": "PostgreSQL is an advanced Object-Relational database management system (DBMS). The image contains the client and server programs that you'll need to create, run, maintain and access a PostgreSQL DBMS server.", "io.k8s.display-name": "PostgreSQL 9.5", "io.openshift.expose-services": "5432:postgresql", "io.openshift.s2i.assemble-user": "26", "io.openshift.s2i.scripts-url": "image:///usr/libexec/s2i", "io.openshift.tags": "database,postgresql,postgresql95,rh-postgresql95", "io.s2i.scripts-url": "image:///usr/libexec/s2i", "maintainer": "SoftwareCollections.org \u003csclorg@redhat.com\u003e", "name": "rhscl/postgresql-95-rhel7", "release": "44", "summary": "PostgreSQL is an advanced Object-Relational database management system", "url": "https://access.redhat.com/containers/#/registry.access.redhat.com/rhscl/postgresql-95-rhel7/images/9.5-44", "usage": "docker run -d --name postgresql_database -e POSTGRESQL_USER=user -e POSTGRESQL_PASSWORD=pass -e POSTGRESQL_DATABASE=db -p 5432:5432 rhscl/postgresql-95-rhel7", "vcs-ref": "66720bffd2fdff4377f92bb574ed17f12f4e0390", "vcs-type": "git", "vendor": "Red Hat, Inc.", "version": "9.5" ...# Realizar o gerenciamento de containers usando comandos como podman e skopeo ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar contêineres**
  - Encontre e recupere imagens de contêiner de um registro remoto
  - Inspecione imagens de contêineres
  - **Execute o gerenciamento de contêineres usando comandos como podman e skopeo**
  - Execute o gerenciamento básico de contêineres, como executar, iniciar, interromper e listar contêineres em execução
  - Execute um serviço dentro de um contêiner
  - Configure um contêiner para iniciar automaticamente como um serviço systemd
  - Anexe armazenamento persistente a um contêiner

Comando podman

Podman (o POD MANager) é uma ferramenta para gerenciar contêineres e imagens, volumes montados nesses contêineres e pods feitos de grupos de contêineres. O Podman é baseado em libpod, uma biblioteca para gerenciamento do ciclo de vida do contêiner que também está contida neste repositório. A biblioteca libpod fornece APIs para gerenciar contêineres, pods, imagens de contêiner e volumes.

```
[root@rhel8 joatham]# man podman
...
NAME
       podman - Simple management tool for pods, containers and images
```

```
SYNOPSIS
       podman [options] command

DESCRIPTION
       Podman (Pod Manager) is a fully featured container engine that is a
       simple daemonless tool.  Podman provides a Docker-CLI comparable
       command line that eases the transition from other container engines and
       allows the management of pods, containers and images.  Simply put:
       alias docker=podman.  Most Podman commands can be run as a regular
       user, without requiring additional privileges.
```

```
COMMANDS
       ------------------------------------------------------
        |Command               | Description                 |
        |----------------------------------------------------|
        |podman-attach(1)      | Attach to a running container.  |
        |----------------------------------------------------|
        |podman-auto-update(1) | Auto update containers      |
        |                      | according to their auto-update  |
        |                      | policy                      |
        |----------------------------------------------------|
        |podman-build(1)       | Build a container image using   |
        |                      | a Containerfile.            |
        |----------------------------------------------------|
        |podman-commit(1)      | Create new image based on the   |
        |                      | changed container.          |
        |----------------------------------------------------|
        |podman-completion(1)  | Generate shell completion   |
        |                      | scripts                     |
        |----------------------------------------------------|
        |podman-container(1)   | Manage containers.          |
        |----------------------------------------------------|
        |podman-cp(1)          | Copy files/folders between a    |
        |                      | container and the local     |
        |                      | filesystem.                 |
        |----------------------------------------------------|
        |podman-create(1)      | Create a new container.     |
        |----------------------------------------------------|
        |podman-diff(1)        | Inspect changes on a container  |
        |                      | or image's filesystem.      |
         `----------------------------------------------------
```

Hands On

- Primeiro precisamos de uma imagem:

```
[root@rhel8 joatham]# podman search ubi | less

INDEX       NAME

     DESCRIPTION                                     STARS       OFFICIAL    AUTOMATED
```

```
redhat.com  registry.access.redhat.com/ubi7/ubi
                                                    The Universal Base Image is designed
     and eng...  0
redhat.com  registry.access.redhat.com/ubi8/ubi
                                                    Provides the latest release of the Red
      Hat U...  0
redhat.com  registry.access.redhat.com/ubi8/ubi-minimal
                                           Provides the latest release of the Minimal R
     ...  0
```

- Baixamos e rodamos a imagem para a máquina

```
[root@rhel8 joatham]# podman run -d registry.access.redhat.com/ubi8/ubi
Trying to pull registry.access.redhat.com/ubi8/ubi:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob 262268b65bd5 [==================================] 79.5MiB / 79.5MiB
Copying blob 06038631a24a done
Copying config 53ce4390f2 done
Writing manifest to image destination
Storing signatures
902f23a38f458ac08938737ceb3adf6be263294f804f9cb4d7617ebb8525d374
```

- Realizamos o gerenciamento de containers

```
[root@rhel8 joatham]# podman ps -a
CONTAINER ID  IMAGE                                           COMMAND     CREATED        STATUS
                         PORTS       NAMES
902f23a38f45  registry.access.redhat.com/ubi8/ubi:latest  /bin/bash   4 minutes ago  Exited
      (0) 4 minutes ago              ecstatic_jones
```

```
[root@rhel8 joatham]# podman run --name teste_ubi registry.access.redhat.com/ubi8/ubi
```

```
[root@rhel8 joatham]# podman ps -a
CONTAINER ID  IMAGE                                           COMMAND     CREATED
     STATUS                      PORTS       NAMES
902f23a38f45  registry.access.redhat.com/ubi8/ubi:latest  /bin/bash   20 minutes ago
     Exited (0) 20 minutes ago              ecstatic_jones
46d935ebb9a8  registry.access.redhat.com/ubi8/ubi:latest  /bin/bash   4 seconds ago
     Exited (0) 4 seconds ago               teste_ubi
```

- Ja vimos que deu certo, agora vamos pegar uma imagem mais consistente, mais utilizável.

```
[root@rhel8 joatham]# podman search httpd | less
```

```
[root@rhel8 joatham]# podman run -d --name servidor_web registry.access.redhat.com/rhscl/
     httpd-24-rhel7
Trying to pull registry.access.redhat.com/rhscl/httpd-24-rhel7:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob b30e8fad55b7 done
Copying blob 71f6d04e5352 done
Copying blob 8a4cee2d3973 done
Copying blob ad62d8acaeb8 done
Copying config ada100d3b5 done
Writing manifest to image destination
Storing signatures
7ec206e689911433ce532b4731acb3353a53d002c40f4dff59743e9f1e0a0a5c
```

```
[root@rhel8 joatham]# podman ps
CONTAINER ID  IMAGE                                                    COMMAND
                  CREATED         STATUS             PORTS       NAMES
7ec206e68991  registry.access.redhat.com/rhscl/httpd-24-rhel7:latest  /usr/bin/run-http...
      58 seconds ago  Up 58 seconds ago              servidor_web
```

Comando skopeo

Utilitário de linha de comando usado para interagir com imagens de contêineres locais e remotos e registros de imagens de contêineres.

- Manual do comando `skokeo`

```
COMMANDS┌┐
        ------------------------------------------------------------------------
        |Command                    | Description                                |╪
        |---------------------------------------------------------------------------
        |skopeo-copy(1)             | Copy an image (manifest, filesystem        |
        |                           | layers, signatures) from one location to   |
        |                           | another.                                   |╪
        |---------------------------------------------------------------------------
        |skopeo-delete(1)           | Mark image-name for deletion.              |╪
        |---------------------------------------------------------------------------
        |skopeo-inspect(1)          | Return low-level information about         |
        |                           | image-name in a registry.                  |╪
        |---------------------------------------------------------------------------
        |skopeo-list-tags(1)        | List the tags for the given                |
        |                           | transport/repository.                      |╪
        |---------------------------------------------------------------------------
        |skopeo-login(1)            | Login to a container registry.             |╪
        |---------------------------------------------------------------------------
```

```
|skopeo-logout(1)             | Logout of a container registry.             |
|-----------------------------------------------------------------------------
|skopeo-manifest-digest(1)    | Compute a manifest digest of manifest-file  |
|                             | and write it to standard output.            |
|-----------------------------------------------------------------------------
|skopeo-standalone-sign(1)    | Sign an image.                              |
|-----------------------------------------------------------------------------
|skopeo-standalone-verify(1)  | Verify an image.                            |
|-----------------------------------------------------------------------------
|skopeo-sync(1)               | Copy images from one or more repositories   |
|                             | to a user specified destination.            |
 `-----------------------------------------------------------------------------
```

- Inspecionando com `Skopeo`

```
[root@rhel8 joatham]# podman images
REPOSITORY                                          TAG         IMAGE ID      CREATED
     SIZE
registry.access.redhat.com/rhscl/httpd-24-rhel7  latest      ada100d3b57e  2 days ago   329
      MB
registry.access.redhat.com/ubi8/ubi              latest      53ce4390f2ad  4 weeks ago  233
      MB
```

```
[root@rhel8 joatham]# skopeo inspect docker://registry.access.redhat.com/rhscl/httpd-24-
     rhel7
{
    "Name": "registry.access.redhat.com/rhscl/httpd-24-rhel7",
    "Digest": "sha256:68ba2e42c882ba30bc995b65b3aa6a3a869178c7ab6ade28e84898f86c685e1d",
    "Created": "2021-10-11T08:11:53.112656Z",
    "DockerVersion": "1.13.1",
    "Labels": {
        "architecture": "x86_64",
        "build-date": "2021-10-11T08:09:56.835819",
        "com.redhat.build-host": "cpt-1003.osbs.prod.upshift.rdu2.redhat.com",
        "com.redhat.component": "httpd24-container",
        "com.redhat.license_terms": "https://www.redhat.com/en/about/red-hat-end-user-
             license-agreements#rhel",
        "description": "Apache httpd 2.4 available as container, is a powerful, efficient,
             and extensible web server. Apache supports a variety of features, many
             implemented as compiled modules which extend the core functionality. These can
              range from server-side programming language support to authentication schemes
             . Virtual hosting allows one Apache installation to serve many different Web
             sites.",
        "distribution-scope": "public",
        "io.k8s.description": "Apache httpd 2.4 available as container, is a powerful,
             efficient, and extensible web server. Apache supports a variety of features,
             many implemented as compiled modules which extend the core functionality.
             These can range from server-side programming language support to
             authentication schemes. Virtual hosting allows one Apache installation to
             serve many different Web sites.",
        "io.k8s.display-name": "Apache httpd 2.4",
        "io.openshift.expose-services": "8080:http,8443:https",
        "io.openshift.s2i.scripts-url": "image:///usr/libexec/s2i",
        "io.openshift.tags": "builder,httpd,httpd24",
        "io.s2i.scripts-url": "image:///usr/libexec/s2i",
```

```
        "maintainer": "SoftwareCollections.org \u003csclorg@redhat.com\u003e",
        "name": "rhscl/httpd-24-rhel7",
        "release": "147.1633939680",
        "summary": "Platform for running Apache httpd 2.4 or building httpd-based
             application",
        "url": "https://access.redhat.com/containers/#/registry.access.redhat.com/rhscl/
             httpd-24-rhel7/images/2.4-147.1633939680",
        "usage": "s2i build https://github.com/sclorg/httpd-container.git --context-dir=
             examples/sample-test-app/ rhscl/httpd-24-rhel7 sample-server",
        "vcs-ref": "1b9deed6782fac7f21124ea2e3d56d75fc1d5fd0",
        "vcs-type": "git",
        "vendor": "Red Hat, Inc.",
        "version": "2.4"
    },
    "Architecture": "amd64",
    "Os": "linux",
    "Layers": [
        "sha256:ad62d8acaeb8e10bb459e0fb98054b6cd0769fe4d0485daf504967c8ffccd2df",
        "sha256:8a4cee2d3973a8b9ccb73fc982adbfef274e95cb2548098e755b1df847aca0de",
        "sha256:71f6d04e5352b855df99a734fa3df8b4ce5c1e73583756a38dae7f0365d48f43",
        "sha256:b30e8fad55b7fb424102f2bf85effaea7d5ea8036c58c248898c0f3558f1ee31"
    ],
    "Env": [
        "PATH=/opt/app-root/src/bin:/opt/app-root/bin:/usr/local/sbin:/usr/local/bin:/usr/
             sbin:/usr/bin:/sbin:/bin",
        "container=oci",
        "SUMMARY=Platform for running Apache httpd 2.4 or building httpd-based application
             ",
        "DESCRIPTION=Apache httpd 2.4 available as container, is a powerful, efficient, and
              extensible web server. Apache supports a variety of features, many
             implemented as compiled modules which extend the core functionality. These can
              range from server-side programming language support to authentication schemes
             . Virtual hosting allows one Apache installation to serve many different Web
             sites.",
        "STI_SCRIPTS_URL=image:///usr/libexec/s2i",
        "STI_SCRIPTS_PATH=/usr/libexec/s2i",
        "APP_ROOT=/opt/app-root",
        "HOME=/opt/app-root/src",
        "PLATFORM=el7",
        "BASH_ENV=/opt/app-root/scl_enable",
        "ENV=/opt/app-root/scl_enable",
        "PROMPT_COMMAND=. /opt/app-root/scl_enable",
        "HTTPD_VERSION=2.4",
        "HTTPD_CONTAINER_SCRIPTS_PATH=/usr/share/container-scripts/httpd/",
        "HTTPD_APP_ROOT=/opt/app-root",
        "HTTPD_CONFIGURATION_PATH=/opt/app-root/etc/httpd.d",
        "HTTPD_MAIN_CONF_PATH=/etc/httpd/conf",
        "HTTPD_MAIN_CONF_MODULES_D_PATH=/etc/httpd/conf.modules.d",
        "HTTPD_MAIN_CONF_D_PATH=/etc/httpd/conf.d",
        "HTTPD_TLS_CERT_PATH=/etc/httpd/tls",
        "HTTPD_VAR_RUN=/var/run/httpd",
        "HTTPD_DATA_PATH=/var/www",
        "HTTPD_DATA_ORIG_PATH=/opt/rh/httpd24/root/var/www",
        "HTTPD_LOG_PATH=/var/log/httpd24",
        "HTTPD_SCL=httpd24"
    ]
}
```


- Copiando uma imagem para um diretório.

```
[root@rhel8 joatham]# skopeo copy docker://registry.access.redhat.com/rhscl/httpd-24-rhel7
     dir:/tmp/container
Getting image source signatures
Checking if image destination supports signatures
Copying blob ad62d8acaeb8 done
Copying blob 8a4cee2d3973 done
Copying blob 71f6d04e5352 done
Copying blob b30e8fad55b7 done
Copying config ada100d3b5 done
Writing manifest to image destination
Storing signatures..
```

```
[root@rhel8 joatham]# ls /tmp/container/
71f6d04e5352b855df99a734fa3df8b4ce5c1e73583756a38dae7f0365d48f43  manifest.json  signature
     -5
8a4cee2d3973a8b9ccb73fc982adbfef274e95cb2548098e755b1df847aca0de  signature-1    signature
     -6
ad62d8acaeb8e10bb459e0fb98054b6cd0769fe4d0485daf504967c8ffccd2df  signature-2    version
ada100d3b57e265696981609562700f251ca8faae5fb36e15268de0b0fbae837  signature-3
b30e8fad55b7fb424102f2bf85effaea7d5ea8036c58c248898c0f3558f1ee31  signature-4
```

> Dica: para excluir uma imagem do armazenamento do contêiner com o Skopeo, use:

```
skopeo delete
```

30

Execute o gerenciamento básico de contêineres, como executar, iniciar, interromper e listar contêineres em execução

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar contêineres**
  - Encontre e recupere imagens de contêiner de um registro remoto
  - Inspecione imagens de contêineres
  - Execute o gerenciamento de contêineres usando comandos como podman e skopeo
  - **Execute o gerenciamento básico de contêineres, como executar, iniciar, interromper e listar contêineres em execução**
  - Execute um serviço dentro de um contêiner
  - Configure um contêiner para iniciar automaticamente como um serviço systemd
  - Anexe armazenamento persistente a um contêiner

Trabalhando com contêiners

Os contêineres representam um processo em execução ou interrompido criado a partir dos arquivos localizados em uma imagem de contêiner descompactada. Você pode usar a ferramenta `Podman` para trabalhar com contêineres.

Podman executando comandos

O comando `podman run` executa um processo em um novo contêiner com base na imagem do contêiner. Se a imagem do contêiner ainda não estiver carregada, o comando `podman run` extrai a imagem e todas as dependências da imagem do repositório da mesma forma que é executado - E antes de iniciar o contêiner a partir dessa imagem. O processo do contêiner possui seu próprio sistema de arquivos, sua própria rede e sua própria árvore de processo isolada. `podman pull image`

O comando `podman run` tem o seguinte formato:

```
podman run [opções] imagem [comando [arg ...]]
```

As opções básicas são:

- –detach (-d): executa o contêiner em segundo plano e imprime a nova ID do contêiner.
- –attach (-a): executa o contêiner no modo de primeiro plano.
- –name (-n): atribui um nome ao contêiner. Se um nome não for atribuído ao contêiner com `--name`, gerará um nome de string aleatório. Isso funciona para contêineres de segundo e primeiro plano.
- –rm: remove automaticamente o contêiner quando ele sair. Observe que o contêiner não será removido se não puder ser criado ou iniciado com êxito.
- –tty (-t): aloca e anexa o pseudoterminal à entrada padrão do contêiner.
- –interactive (-i): para processos interativos, use `-i` e em `-t` conjunto para alocar um terminal para o processo de contêiner.

Listagem de contêineres

Use o comando `podman ps` para listar os contêineres em execução no sistema.

Pré-requisitos

- A ferramenta Podman está instalada.

```
[root@rhel8 joatham]# yum module install -y container-tools
```

Procedimento

- Execute o contêiner com base na registry.redhat.io/rhel8/rsyslogimagem:

```
[root@rhel8 joatham]# podman run -d registry.redhat.io/rhel8/rsyslog
```

- Liste todos os contêineres:

Para listar todos os contêineres em execução:

```
[root@rhel8 joatham]# podman ps
CONTAINER ID IMAGE              COMMAND         CREATED       STATUS            PORTS NAMES
74b1da000a11 rhel8/rsyslog /bin/rsyslog.sh 2 minutes ago Up About a minute
     musing_brown
```

- Para listar todos os contêineres que estão em execução ou parados, utilize:

```
[root@rhel8 joatham]# podman ps -a
CONTAINER ID IMAGE         COMMAND    CREATED    STATUS                PORTS NAMES     IS
     INFRA
d65aecc325a4 ubi8/ubi      /bin/bash  3 secs ago Exited (0) 5 secs ago peaceful_hopper
     false
74b1da000a11 rhel8/rsyslog rsyslog.sh 2 mins ago Up About a minute     musing_brown
     false
```

Startando Contêineres

Se você executar o contêiner e, em seguida, interrompê-lo em vez de removê-lo, o contêiner será armazenado em seu sistema local pronto para ser executado novamente. Você pode usar o comando `podman start` para executar novamente os contêineres. Você pode especificar os contêineres por seu ID ou nome de contêiner.

Hands On

Inicie o contêiner teste:

- No modo não interativo:

```
[root@rhel8 joatham]# podman start teste
```

Alternativamente, você pode usar `podman start 1984555a2c27`.

- No modo interativo, use as opções `-a( --attach)` e `-t( --interactive)` para trabalhar com o shell bash do contêiner:

```
[root@rhel8 joatham]# podman start -a -i teste
```

Alternativamente, você pode usar o comando `podman start -a -i 1984555a2c27`.

Digite `exit` para sair do contêiner e retornar ao host:

```
[root@rhel8 joatham]# exit
```

Parando contêineres

Use o comando `podman stop` para parar um contêiner em execução. Você pode especificar os contêineres por seu ID ou nome de contêiner.

Hands On

Pare o contêiner teste:

- Usando o nome do contêiner:

```
[root@rhel8 joatham]# podman para teste
```

- Usando o ID do contêiner:

```
[root@rhel8 joatham]# podman stop 1984555a2c27
```

Para interromper um contêiner em execução anexado a uma sessão de terminal, você pode inserir o comando `exit` dentro do contêiner.

O comando `podman stop` envia um sinal `SIGTERM` para encerrar um contêiner em execução. Se o contêiner não parar após um período definido (10 segundos por padrão), o Podman envia um sinal `SIGKILL`.

Você também pode usar o comando `podman kill` para matar um contêiner (SIGKILL) ou enviar um sinal diferente para um contêiner. Aqui está um exemplo de envio de um sinal `SIGHUP` para um contêiner (se suportado pelo aplicativo, um `SIGHUP` faz com que o aplicativo releia seus arquivos de configuração):

```
[root@rhel8 joatham]# podman kill --signal="SIGHUP" 74b1da000a11
74b1da000a114015886c557deec8bed9dfb80c888097aa83f30ca4074ff55fb2
```

Removendo contêiners

Use o comando `podman rm` para remover containers. Você pode especificar contêineres com o ID ou nome do contêiner.

Hands On

- Liste todos os contêineres, em execução ou parados:

```
[root@rhel8 joatham]# podman ps -a
CONTAINER ID IMAGE         COMMAND    CREATED    STATUS                PORTS NAMES     IS
     INFRA
d65aecc325a4 ubi8/ubi      /bin/bash  3 secs ago Exited (0) 5 secs ago teste
     false
74b1da000a11 rhel8/rsyslog rsyslog.sh 2 mins ago Up About a minute     musing_brown
     false
```

- Remova os contêiners:

Para remover o contêiner teste:

```
[root@rhel8 joatham]# podman rm teste
```

Observe que o contêiner teste estava com status `exit`, o que significa que foi interrompido e pode ser removido imediatamente.

Para remover o contêiner teste, primeiro pare e, em seguida, remova-o:

```
[root@rhel8 joatham]# podman stop teste
```

shell [root@rhel8 joatham]# podman rm teste# Executar um serviço em um container ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar contêineres**
  - **Encontre e recupere imagens de contêiner de um registro remoto**
  - Inspecione imagens de contêineres
  - Execute o gerenciamento de contêineres usando comandos como podman e skopeo
  - Execute o gerenciamento básico de contêineres, como executar, iniciar, interromper e listar contêineres em execução
  - **Execute um serviço dentro de um contêiner**
  - Configure um contêiner para iniciar automaticamente como um serviço systemd
  - Anexe armazenamento persistente a um contêiner

Hands On

- Vamos escolher um serviço para realizar os testes! Neste caso, como de costume, escolheremos um servidor WEB.

```
[root@rhel8 joatham]# podman search apache | grep -i rhel7
redhat.com  registry.access.redhat.com/rhscl/httpd-24-rhel7
                                                Apache HTTP 2.4 Server
                                 0
redhat.com  registry.access.redhat.com/rhscl/php-70-rhel7
                                                  PHP 7.0 platform for building and running
     ap...  0
redhat.com  registry.access.redhat.com/openshift3/php-55-rhel7
                                               PHP 5.5 platform for building and running ap...
       0
redhat.com  registry.access.redhat.com/rhscl/php-71-rhel7
                                                  PHP 7.1 available as container is a base
     pla...  0
```

- Execute a imagem em uma porta específica:

```
[root@rhel8 joatham]# podman run --name Servidor_web_1 -dit -p 14000:8080 registry.access.
     redhat.com/rhscl/httpd-24-rhel7
Trying to pull registry.access.redhat.com/rhscl/httpd-24-rhel7:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob 71f6d04e5352 done
Copying blob 8a4cee2d3973 done
```

```
Copying blob ad62d8acaeb8 done
Copying blob b30e8fad55b7 done
Copying config ada100d3b5 done
Writing manifest to image destination
Storing signatures
639e4d96f4c328295df72842cbe7b9a49d55d6c2b5cd4e351e4e811ffa54062c
```

```
[root@rhel8 joatham]# podman ps
CONTAINER ID  IMAGE                                                    COMMAND
                  CREATED        STATUS            PORTS                     NAMES
639e4d96f4c3  registry.access.redhat.com/rhscl/httpd-24-rhel7:latest  /usr/bin/run-http...
      6 minutes ago  Up 6 minutes ago  0.0.0.0:14000->8080/tcp  Servidor_web_1
```

> Teste no seu navegador: `http://localhost:14000`

- Execute o `/bin/bash` dentro do container:

```
[root@rhel8 joatham]# podman exec -it Servidor_web_1 /bin/bash
bash-4.2$ whoami
default
bash-4.2$
```

```
bash-4.2$ echo "Nao esquece de me mandar um salve pro @joatham.pedro no @4linux..." > /var/
     www/html/4linux.txt
bash-4.2$
```

> Teste no seu navegador: `http://localhost:14000/4linux.txt`

- Pare o serviço com o comando abaixo:

```
[root@rhel8 joatham]# podman stop Servidor_web_1
Servidor_web_1
```

> Teste no seu navegador: `http://localhost:14000`

ou

```
[root@rhel8 joatham]# curl localhost:14000
curl: (7) Failed to connect to localhost port 14000: Conexão recusada
```

31

Configurar um contêiner para ser iniciado automaticamente como um serviço do systemd

Pontos de estudo para o exame

Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar contêineres**
  - Encontre e recupere imagens de contêiner de um registro remoto
  - Inspecione imagens de contêineres
  - Execute o gerenciamento de contêineres usando comandos como podman e skopeo
  - Execute o gerenciamento básico de contêineres, como executar, iniciar, interromper e listar contêineres em execução
  - Execute um serviço dentro de um contêiner
  - **Configure um contêiner para iniciar automaticamente como um serviço systemd**
  - Anexe armazenamento persistente a um contêiner

Hands On

- Antes de começar, vamos dar uma olhada no manual do comando `podman-generate`:

```
podman-generate(1)                General Commands Manual                podman-
     generate(1)

NAME
       podman-generate - Generate structured data based on containers, pods or volumes

SYNOPSIS
       podman generate subcommand

DESCRIPTION
       The generate command will create structured output (like YAML) based on a container,
             pod
       or volume.

COMMANDS┌┌┐
       ----------------------------------------------------------
        |Command  | Man Page                   | Description              |┼┼|
        |---------------------------------------------------------------
        |kube     | podman-generate-kube(1)    | Generate Kubernetes YAML  |
        |         |                            | based on containers,      |
        |         |                            | pods or volumes.          |┼┼|
        |---------------------------------------------------------------
        |systemd  | podman-generate-systemd(1) | Generate systemd unit     |
        |         |                            | file(s) for a container   |
        |         |                            | or pod.                   |└└|
         `---------------------------------------------------------------

SEE ALSO
       podman, podman-pod, podman-container
```

```
[joatham@rhel8 ~]$ podman generate systemd Servidor_web_1
# container-090e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194.service
# autogenerated by Podman 3.2.3
# Wed Oct 13 21:20:01 -03 2021

[Unit]
Description=Podman container-090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194.service
Documentation=man:podman-generate-systemd(1)
Wants=network.target
After=network-online.target
RequiresMountsFor=/run/containers/storage

[Service]
Environment=PODMAN_SYSTEMD_UNIT=%n
Restart=on-failure
TimeoutStopSec=70
ExecStart=/usr/bin/podman start 090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194
ExecStop=/usr/bin/podman stop -t 10 090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194
ExecStopPost=/usr/bin/podman stop -t 10 090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194
```

```
PIDFile=/run/containers/storage/overlay-containers/090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194/userdata/conmon.pid
Type=forking

[Install]
WantedBy=multi-user.target default.target
```

Já podemos perceber que o comando `podman-generate` nos proporcionou todo o arquivo `[unit]` que o `Systemd` precisa. Assim, seria interessante ajustarmos essa configuração para o usuário que terá acesso ao serviço na inicialização. Portanto, vamos rodar o comando no diretório correto para que o `Systemd` o reconheça. Acompanhe os comandos a seguir:

- Primeiro, vamos criar a pasta para agrupar o arquivo `.service`

```
[joatham@rhel8 ~]$ mkdir -p /home/joatham/.config/systemd/joatham
```

- Em seguida, executamos o comando `podman generate` gerando um arquivo dentro deste diretório:

```
[joatham@rhel8 ~]$ podman generate systemd Servidor_web_1 > Servidor_web_1.service
```

```
[joatham@rhel8 ~]$ cat Servidor_web_1.service
# container-090e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194.service
# autogenerated by Podman 3.2.3
# Wed Oct 13 21:30:29 -03 2021

[Unit]
Description=Podman container-090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194.service
Documentation=man:podman-generate-systemd(1)
Wants=network.target
After=network-online.target
RequiresMountsFor=/run/containers/storage

[Service]
Environment=PODMAN_SYSTEMD_UNIT=%n
Restart=on-failure
TimeoutStopSec=70
ExecStart=/usr/bin/podman start 090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194
ExecStop=/usr/bin/podman stop -t 10 090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194
ExecStopPost=/usr/bin/podman stop -t 10 090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194
PIDFile=/run/containers/storage/overlay-containers/090
     e67643182f51e940590c759b69f57bae52432f157222005601f8560e77194/userdata/conmon.pid
Type=forking
```

```
[Install]
WantedBy=multi-user.target default.target
```

- Verifique o status do serviço com o comando `systemctl`

```
[joatham@rhel8 user]$ systemctl --user status servidor_web_1.service
Unit servidor_web_1.service could not be found.
```

> Retornou ERRO!!! Você precisa dar um reload no Systemd.

```
[joatham@rhel8 ~]$ systemctl --user daemon-reload
```

- Agora, verifique o status do serviço com o `systemctl`:

```
[joatham@rhel8 ~]$ systemctl --user status servidor_web_1.service●
 servidor_web_1.service - Podman container-2
      e5d2faad2caf32589c47b07bb47f92abe29b5bb8bdb8e0d2d0c5>
   Loaded: loaded (/home/joatham/.config/systemd/user/servidor_web_1.service; disabled;
        vendor pr>
   Active: inactive (dead)
     Docs: man:podman-generate-systemd(1)
```

- Por fim, como qualquer serviço SystemD, execute o comando `enable`:

```
[joatham@rhel8 ~]$ systemctl --user enable servidor_web_1.service --now
Created symlink /home/joatham/.config/systemd/user/multi-user.target.wants/servidor_web_1.
     service → /home/joatham/.config/systemd/user/servidor_web_1.service.
Created symlink /home/joatham/.config/systemd/user/default.target.wants/servidor_web_1.
     service → /home/joatham/.config/systemd/user/servidor_web_1.service.
Job for servidor_web_1.service failed because the service did not take the steps required
     by its unit configuration.
See "systemctl --user status servidor_web_1.service" and "journalctl --user -xe" for
     details.
```

```
[joatham@rhel8 ~]$ podman ps
CONTAINER ID  IMAGE                                                    COMMAND
                  CREATED         STATUS             PORTS                     NAMES
2e5d2faad2ca  registry.access.redhat.com/rhscl/httpd-24-rhel7:latest  /usr/bin/run-http...
      17 minutes ago  Up 15 seconds ago  0.0.0.0:14000->8080/tcp  Servidor_web_1
```

> Teste no seu navegador: `http://localhost:14000/4linux.txt`# Anexar armazenamento persistente a um container ## Pontos de estudo para o exame Os candidatos ao exame RHCSA devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos em várias categorias:

- **Gerenciar contêineres**
  - Encontre e recupere imagens de contêiner de um registro remoto
  - Inspecione imagens de contêineres
  - Execute o gerenciamento de contêineres usando comandos como podman e skopeo
  - Execute o gerenciamento básico de contêineres, como executar, iniciar, interromper e listar contêineres em execução
  - Execute um serviço dentro de um contêiner
  - Configure um contêiner para iniciar automaticamente como um serviço systemd
  - **Anexe armazenamento persistente a um contêiner**

Introdução

Vamos começar tudo de novo para que possamos entender todo o processo:

1. Deletar todas as imagens;
2. Procurá-las;
3. Baixar para nossa máquina;
4. Executarmos com foco no armazenamento persistente!

Hands On

- Deletar todas as imagens:

```
[root@rhel8 joatham]# podman rmi a --force
Untagged: registry.access.redhat.com/rhscl/httpd-24-rhel7:latest
Deleted: ada100d3b57e265696981609562700f251ca8faae5fb36e15268de0b0fbae837
```

```
[root@rhel8 joatham]# podman ps
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
```

```
[root@rhel8 joatham]# podman images
REPOSITORY  TAG         IMAGE ID    CREATED     SIZE
```

- Procurar novamente dentro dos repositórios da RHEL a imagem do Apache:

```
[root@rhel8 joatham]# podman search httpd
INDEX       NAME
     DESCRIPTION                                     STARS       OFFICIAL    AUTOMATED
redhat.com  registry.access.redhat.com/rhscl/httpd-24-rhel7
     Apache HTTP 2.4 Server                          0
redhat.com  registry.access.redhat.com/cloudforms46-beta/cfme-openshift-httpd
     CloudForms is a management and automation pl...  0
redhat.com  registry.access.redhat.com/cloudforms46/cfme-openshift-httpd
     Web Server image for a multi-pod Red Hat® C...   0
```

- Realizar o pull da imagem encontrada:

```
[root@rhel8 joatham]# podman pull registry.access.redhat.com/rhscl/httpd-24-rhel7
Trying to pull registry.access.redhat.com/rhscl/httpd-24-rhel7:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob ea1bb0040f98 done
Copying blob 0bcb99761896 done
Copying blob c65299f28473 done
Copying blob 86a352f9781b done
Copying config 1741df917e done
Writing manifest to image destination
Storing signatures
1741df917e2f10681f12f4fa2bca876a8ca5db40802713abd3d4aad3c38f5143
```

- Testar o armazenamento persistente:

Sistemas de etiquetagem como o SELinux requerem que etiquetas adequadas sejam colocadas no conteúdo do volume montado em um contêiner. Sem um rótulo, o sistema de segurança pode impedir que os processos em execução dentro do contêiner usem o conteúdo. Por padrão, o Podman não altera os rótulos definidos pelo sistema operacional.

```
[root@rhel8 joatham]# mkdir httpd
[root@rhel8 joatham]# echo "testando armazenamento persistente 4linux 1" > /httpd/4linux1.
     txt
```

```
[root@rhel8 joatham]# podman run --name 4linux_web_1 -dit -p 7000:8080 -v /home/joatham/
     httpd:/var/www/html:z registry.access.redhat.com/rhscl/httpd-24-rhel7
f414edc00df14bbf3d1691fbbb0860e1db03675d666135b306a027662ffb6c7c
```

Para alterar um rótulo no contexto do contêiner, você pode adicionar um dos dois sufixos

`:z` ou `:Z` à montagem do volume. Esses sufixos dizem ao `Podman` para rotular novamente os objetos de arquivo nos volumes compartilhados. A opção `z` diz ao `Podman` que dois contêineres compartilham o conteúdo do volume. Como resultado, o `Podman` rotula o conteúdo com um rótulo de conteúdo compartilhado. Rótulos de volume compartilhado permitem que todos os contêineres leiam/gravem conteúdo. Em outras palavras, a opção `Z` diz ao `Podman` para rotular o conteúdo com um rótulo privado não compartilhado.

```
[root@rhel8 joatham]# podman ps
CONTAINER ID  IMAGE                                                COMMAND
              CREATED         STATUS             PORTS                   NAMES
f414edc00df1  registry.access.redhat.com/rhscl/httpd-24-rhel7:latest  /usr/bin/run-http...
      50 seconds ago  Up 50 seconds ago  0.0.0.0:7000->8080/tcp  4linux_web_1
```

> Teste em seu navegador: http://localhost:7000/4linux1.txt

- Verifique os contextos do **`SELinux`**:

```
[root@rhel8 joatham]# ls -lZ
total 588800
drwxr-xr-x. 2 joatham joatham unconfined_u:object_r:user_home_t:s0          6 out  1 17:31
     'Área de trabalho'
-rw-r--r--. 1 root    root    unconfined_u:object_r:user_home_t:s0  602931200 out 26  2020
      boot.iso
drwxr-xr-x. 2 joatham joatham unconfined_u:object_r:user_home_t:s0          6 out  1 17:31
      Documentos
drwxr-xr-x. 2 joatham joatham unconfined_u:object_r:user_home_t:s0          6 out  1 17:31
      Downloads
drwxr-xr-x. 2 root    root    system_u:object_r:container_file_t:s0        25 out 14 15:23
      httpd
drwxr-xr-x. 2 joatham joatham unconfined_u:object_r:user_home_t:s0          6 out  1 17:31
      Imagens
drwxr-xr-x. 2 joatham joatham unconfined_u:object_r:user_home_t:s0          6 out  1 17:31
      Modelos
drwxr-xr-x. 2 joatham joatham unconfined_u:object_r:user_home_t:s0          6 out  1 17:31
      Música
drwxr-xr-x. 2 joatham joatham unconfined_u:object_r:user_home_t:s0          6 out  1 17:31
      Público
drwxr-xr-x. 2 joatham joatham unconfined_u:object_r:user_home_t:s0          6 out  1 17:31
      Vídeos
```

- Crie uma nova pasta e testando seus contextos:

```
[root@rhel8 joatham]# mkdir httpd2
[root@rhel8 joatham]# ls -lZ
total 588800
drwxr-xr-x. 2 root    root    system_u:object_r:container_file_t:s0        25 out 14 15:23
      httpd
```

```
drwxr-xr-x. 2 root    root    unconfined_u:object_r:user_home_t:s0          6 out 14 17:08
      httpd2
```

- Vamos refazer o mesmo teste, só que desta vez sem a flag `z` para vermos o que acontece com o conflito de contextos do `Selinux`:

```
[root@rhel8 joatham]# echo "testando armazenamento persistente 4linux 2" > httpd2/4linux2.
     txt

[root@rhel8 joatham]# ls -lRZ httpd*
httpd:
total 4
-rw-r--r--. 1 root root system_u:object_r:container_file_t:s0 44 out 14 15:23 4linux1.txt

httpd2:
total 4
-rw-r--r--. 1 root root unconfined_u:object_r:user_home_t:s0 44 out 14 17:11 4linux2.txt
```

- Executaremos nosso contêiner sem aplicarmos o contexto do Selinux (flag `z`)

```
[root@rhel8 joatham]# podman run --name 4linux_web_2 -dit -p 7001:8080 -v /home/joatham/
     httpd2/:/var/www/html/ registry.access.redhat.com/rhscl/httpd-24-rhel7
0ccc068c7ba03174cf53706040bf9e97f13594878323f994e9584a9cfbe4bd5d
```

```
[root@rhel8 joatham]# podman ps
CONTAINER ID  IMAGE                                                    COMMAND
                  CREATED         STATUS             PORTS                   NAMES
f414edc00df1  registry.access.redhat.com/rhscl/httpd-24-rhel7:latest  /usr/bin/run-http...
      2 hours ago     Up 2 hours ago     0.0.0.0:7000->8080/tcp  4linux_web_1
0ccc068c7ba0  registry.access.redhat.com/rhscl/httpd-24-rhel7:latest  /usr/bin/run-http...
      23 seconds ago  Up 23 seconds ago  0.0.0.0:7001->8080/tcp  4linux_web_2
```

> Teste em seu navegador: http://localhost:7000/4linux2.txt

Uma outra forma de constatar este fato seria entrando dentro do contêiner com o comando `podman exec`.

- Bash do 4linux_web_2

```
[root@rhel8 joatham]# podman exec -it 4linux_web_2 /bin/bash
bash-4.2$ ls -lZ /var/www/
```

```
drwxr-xr-x. default root system_u:object_r:container_file_t:s0:c88,c916 cgi-bin
drwxr-xr-x. root    root unconfined_u:object_r:user_home_t:s0 html
bash-4.2$
```

- Bash do 4linux_web_1

```
[root@rhel8 joatham]# podman exec -it 4linux_web_1 /bin/bash
bash-4.2$ ls -lZ /var/www/
drwxr-xr-x. default root system_u:object_r:container_file_t:s0:c732,c946 cgi-bin
drwxr-xr-x. root    root system_u:object_r:container_file_t:s0 html
bash-4.2$
```

`shell bash-4.2$ cd /var/www/html/ bash-4.2$ ls -lZ -rw-r--r--. root root system_u:object_r:container_file_t :s0 4linux1.txt bash-4.2$`# Pesquisar arquivos | Comparar e manipular o conteúdo de arquivos ## Pontos de estudo para o exame Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Comandos essenciais**
  - Faça login nos consoles de modo gráfico e de texto locais e remotos
  - **Pesquise arquivos**
  - Avalie e compare os recursos e opções básicas do sistema de arquivos
  - **Compare e manipule o conteúdo do arquivo**
  - Use o redirecionamento de entrada-saída (por exemplo,>, », |, 2>)
  - Analise o texto usando o regular básico Expressões
  - Arquive, faça backup, compacte e descompacte arquivos
  - Crie, exclua, copie e mova arquivos e diretórios
  - Crie e gerencie links físicos e virtuais
  - Liste, defina e altere permissões de arquivo padrão
  - Leia e use a documentação do sistema
  - Gerencie acesso à raiz root

Introdução

Você já precisou encontrar algum arquivo ou diretório em uma distribuição Linux e não soube por onde começar? Aqui, te apresentamos a solução para esse tipo de problema, sua solução chama-se `find`.

hands On

A sintaxe mais comum ao utilizar o comando `find` é:

```
find <diretório de origem> <opções> <termo de busca>
```

Onde:

- find –> Comando de busca
- -> Diretório onde a pesquisa será iniciada
- -> Opções que serão aplicadas na consulta
- -> Nome do arquivo ou diretório a ser encontrado

Para realizar uma busca por todas as pastas e diretórios do sistema, o caminho de origem deve ser definido como /.

Pesquise em `/etc` todos os arquivos `/` diretórios com `host` em seus nomes. * é um curinga.

```
[root@rhel8 joatham]# find / -name "\*host*"
```

Pesquise a partir da posição atual todos os arquivos com permissões `777` e depois remova-os:

```
find . -perm 777 -exec rm -f '{}' \;
```

- -exec usa o resultado de encontrar para fazer algo
- {} será substituído pelo resultado de encontrar
- O comando do `exec` deve estar entre `-exec` e `\;`.
- ; é tratado como caractere de fim de comando no shell bash. Para isso, deve-se escapar com `\`. Se escapar, será interpretado pelo `find` e não por bash shell.

Alguns parâmetros aceitam o valor `n` com `+` ou `-` na frente. O significado é:

- +n - para maior que n
- -n - por menos de n
- n - por exatamente n

Pesquise no diretório `/etc` todos os arquivos com tamanho inferior a 100 kilobytes

```
[root@rhel8 joatham]# find /etc -size -100k
```

Pesquisa começando da posição atual, descendo no máximo três níveis de diretórios, arquivos com tamanho maior de 2 megabytes

```
[root@rhel8 joatham]# find . -maxdepth 3 -type f -size +2M
```

A flag `-o` é usado para combinar duas condições. `\` é uma fuga para evitar que (ou) seja interpretado por bash shell

```
[root@rhel8 joatham]# find . \( -name name1 -o -name name2 \)
```

Encontre todos os arquivos que têm o mesmo i-node do arquivo:

```
[root@rhel8 joatham]# find . -samearquivo arquivo
```

Ele mostrará todos os arquivos que não pertencem ao proprietário do usuário. `!` significa negação, mas deve ser escapado por `\` para não ser interpretado pelo shell bash

```
[root@rhel8 joatham]# find . \! -user owner
```

Pesquisar nome ignorando maiúsculas e minúsculas:

```
[root@rhel8 joatham]# find . -iname name
```

Encontre todos os arquivos com permissões iguais a `222`. Por exemplo, apenas o arquivo com permissões `222` será mostrado:

```
[root@rhel8 joatham]# find . -perm 222
```

Encontre todos os arquivos com pelo menos permissões 222. Por exemplo, 777 correspondências como válidas:

```
[root@rhel8 joatham]# find . -perm -222
```

Encontre todos os arquivos com write para owner ou write para group ou write para others (pelo menos um):

```
[root@rhel8 joatham]# find . -perm /222
```

Encontre todos os arquivos com pelo menos permissão de gravação para o grupo:

```
[root@rhel8 joatham]# find . -perm -g=w
```

Mostrar todos os arquivos acessados há pelo menos dois dias (mais de 24 horas).

```
[root@rhel8 joatham]# find . -atime +1
```

Pesquisar arquivos | Compare e manipule o conteúdo do arquivo

Nos sistemas operacionais do tipo Unix, o comando `diff` analisa dois arquivos e imprime as linhas diferentes. Em essência, ele fornece um conjunto de instruções sobre como alterar um arquivo para torná-lo idêntico ao segundo arquivo.

Compare o arquivo1 e o arquivo2

```
[root@rhel8 joatham]# diff arquivo1 arquivo2
```

Compare o arquivo 1 e o arquivo 2 com a saída em duas colunas

```
[root@rhel8 joatham]# diff -y arquivo1 arquivo2
```

Remover linhas consecutivas iguais

```
[root@rhel8 joatham]# uniq arquivo
```

Remova linhas consecutivas iguais comparando apenas os dois primeiros caracteres

```
[root@rhel8 joatham]# uniq -w 2 arquivo
```

Remova linhas consecutivas iguais e mostre o número de ocorrências

```
[root@rhel8 joatham]# uniq -c arquivo
```

Ordenar o conteúdo do arquivo

```
[root@rhel8 joatham]# sort arquivo
```

Ordene o conteúdo do arquivo usando como referência a segunda palavra

```
[root@rhel8 joatham]# sort -k 2 arquivo
```

Comando `cut`

Imprime a primeira palavra de cada linha. O delimitador será o espaço:

```
c[root@rhel8 joatham]# cut -d ' ' -f 1 arquivo
```

Imprime a primeira e a terceira palavra de cada linha. O delimitador será o espaço:

```
[root@rhel8 joatham]# cut -d ' ' -f 1,3 arquivo
```

Imprimir o conteúdo do arquivo

```
[root@rhel8 joatham]# cat arquivo
```

Imprimir as últimas 10 linhas do arquivo

```
[root@rhel8 joatham]# tail arquivo
```

Imprime as últimas 5 linhas do arquivo

```
[root@rhel8 joatham]# tail -n 5 arquivo
```

Imprima as últimas 10 linhas do arquivo e acrescente. Útil para monitorar arquivos de log

```
[root@rhel8 joatham]# tail -f arquivo
```

Imprime as primeiras 10 linhas do arquivo

```
[root@rhel8 joatham]# head arquivo
```

Imprime as 2 primeiras linhas do arquivo

```
[root@rhel8 joatham]# head -n 2 arquivo
```

Traduzir conjunto de caracteres um para conjunto de caracteres 2

```
[root@rhel8 joatham]# tr SET1 SET2
```

Ele substituirá todas as ocorrências de teste com sub

```
[root@rhel8 joatham]# cat arquivo | tr test sub
```

Substituirá todas as ocorrências consecutivas de espaço por um espaço

```
[root@rhel8 joatham]# cat arquivo | tr -s ' '
```

Imprime o tipo de nome do arquivo

`shell [root@rhel8 joatham]# file namearquivo`# Diagnosticar e gerenciar processos | Alterar os parâmetros de tempo de execução do kernel, persistentes e não persistentes ## Pontos de estudo para o exame Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Operação de Sistemas de Funcionamento**
  - Inicialize, reinicialize e desligue um sistema com segurança
  - Inicialize ou altere o sistema em modos operacionais diferentes
  - Instale, configure e solucione problemas de bootloaders
  - **Diagnostique e gerencie processos**
  - Localize e analise arquivos de log do sistema
  - Agende tarefas para serem executadas em uma data e hora definidas
  - Verifique a conclusão dos trabalhos agendados
  - Atualize software para fornecer a funcionalidade e segurança necessárias
  - Verifique a integridade e disponibilidade dos recursos
  - Verifique a integridade e disponibilidade dos processos-chave
  - **Altere os parâmetros de tempo de execução do kernel, persistentes e não persistentes**
  - Use scripts para automatizar as tarefas de manutenção do sistema
  - Gerencie o processo de inicialização e serviços (Configuração nos serviços)
  - Liste e identifique o arquivo SELinux / AppArmor e os contextos do processo
  - Gerencie software
  - Identifique o componente de uma distribuição Linux ao qual um arquivo pertence

Introdução

`Sysstat` é realmente uma ferramenta útil que vem com vários utilitários para monitorar os recursos do sistema, seu desempenho e atividades de uso. Número de utilitários que todos nós usamos em nossas bases diárias vem com o pacote `sysstat`. Ele também fornece a ferramenta que pode ser agendada usando o `cron` para coletar todos os dados de desempenho e atividade. A seguir está a lista de ferramentas incluídas nos pacotes `sysstat`.

1. **iostat**: relata todas as estatísticas sobre sua CPU e estatísticas de E/S para dispositivos de E/S.
2. **mpstat**: detalhes sobre CPUs (individuais ou combinadas).
3. **pidstat**: estatísticas sobre processos/tarefas em execução, CPU, memória etc.
4. **sar**: salvar e relatar detalhes sobre diferentes recursos (CPU, Memória, IO, Rede, kernel etc.).
5. **sadc**: coletor de dados de atividade do sistema, usado para coletar dados no backend

para sar.

6. **sa1**: busca e armazena dados binários no arquivo de dados sadc. Isso é usado com sadc.
7. **sa2**: relatório diário de resumos para ser usado com o sar.
8. **Sadf**: usado para exibir dados gerados pelo sar em diferentes formatos (CSV ou XML).
9. **Sysstat**: página de manual do utilitário sysstat.
10. **nfsiostat-sysstat**: estatísticas de E/S para NFS.
11. **cifsiostat**: estatísticas para CIFS.

hands On

Comando `mpstat`

```
[root@localhost joatham]# yum -y install sysstat
```

```
[root@localhost joatham]# mpstat -P ALL -u 2 3
```

Estatísticas de uso da CPU.

- **-P** Indique o número do processador para o qual as estatísticas devem ser relatadas, TODAS para todas as cpu.
- **-u** Relatório de utilização da CPU.
- **2 3** Exiba três relatórios em intervalos de dois segundos.

Comando `ps`

Processos dos quais sou proprietário

```
[root@localhost joatham]# ps
```

Todos os processos.

```
[root@localhost joatham]# ps aux
```

Vai imprimir:

- usuário - Usuário que possui o processo

- pid - ID de processo do processo
- % cpu - É o tempo de CPU usado dividido pelo tempo de execução do processo
- % mem - Proporção do tamanho do conjunto residente do processo para a memória física na máquina
- VSZ (memória virtual) - Uso de memória virtual de todo o processo (em KiB)
- RSS (memória residente) - Tamanho do conjunto residente, a memória física não trocada que uma tarefa usou (em KiB)
- tty - No qual o processo está sendo executado.
- ? - significa que não está conectado a um tty
- stat - Estado do processo
- start - Hora ou data de início do processo
- time - Tempo de CPU cumulativo
- comando - comando com todos os seus argumentos

> Aqueles dentro de [] são processos do sistema ou thread do kernel

```
[root@localhost joatham]# ps -eo pid,ppid,cmd,%cpu,%mem --sort=-%cpu
```

- **-e** - Mostrar o mesmo resultado de aux
- **-o** - Escolheu colunas para mostrar
- **–sort** - Classificar pelo parâmetro fornecido
- **ppid** - id do processo pai

```
[root@localhost joatham]# ps -e -o pid,args --forest
```

- **–forest** - Mostrar uma visão gráfica da árvore de processos

Alterar os parâmetros de tempo de execução do kernel, persistentes

O kernel, em se tratando de sistemas operacionais, é o núcleo e componente mais importante da maioria dos computadores. Basicamente, serve de ponte entre os aplicativos e o processamento real de dados feito a nível de hardware. É ele o responsável por gerenciar os recursos do sistema, podendo oferecer uma camada de abstração de nível mais baixo para os recursos, como processadores e dispositivos de entrada/saída, que os softwares aplicativos devem controlar para realizar sua função. Com o GNU/Linux não é diferente. O núcleo Linux (Linux Kernel)

forma a estrutura do sistema operacional GNU/Linux.

Como é de se esperar, o kernel possui diversos parâmetros configurados que definirão as características do seu sistema, controle de dispositivos, módulos, drivers, etc. Por vezes faz-se necessário alterar algum parâmetro do kernel para alguma tarefa ou rotina específica, portanto que tal ganhar tempo e alterar um ou mais parâmetros do kernel on the fly?!

O comando `sysctl` pode ajudar nesta tarefa. Ele ajuda a configurar os parâmetros do kernel em tempo de execução.

Para listar os atuais parâmetros de seu kernel, digite:

```
[root@localhost joatham]# sysctl -a | head
abi.vsyscall32 = 1
crypto.fips_enabled = 0
debug.exception-trace = 1
debug.kprobes-optimization = 1
dev.cdrom.autoclose = 1
dev.cdrom.autoeject = 0
dev.cdrom.check_media = 0
dev.cdrom.debug = 0
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17
dev.cdrom.info =
```

O retorno deste comando é bastante extenso, portanto, colei aqui apenas algumas linhas aleatórias de meu resultado.

Para alterar temporariamente um parâmetro, utilize o parâmetro `-w` do `sysctl`, indicando a variável que deseja alterar e o novo valor que será utilizado para ela:

```
[root@localhost joatham]#  sysctl -w {nome-da-variável=valor}
```

No caso acima, a(s) alteração(ões) será(ão) perdida(s) após a reinicialização do sistema.

Caso deseje realizar alterações permanentes, edite o arquivo `/etc/sysctl.conf` e, em seguida, aplique suas modificações com o parâmetro `-p` do `sysctl`.

[root@localhost joatham]# sysctl -p

Desta forma, após a reinicialização suas modificações permanecerão ativas.

32

Gerenciar ambiente de usuário de modelo | Configurar PAM

Pontos de estudo para o exame

Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Gerenciamento de usuários e grupos**
  - Criar, excluir e modificar contas de usuários locais
  - Criar, excluir e modificar grupos locais e associações de grupos
  - Gerenciar perfis de ambiente de todo o sistema
  - **Gerenciar ambiente de usuário de modelo**
  - Configurar limites de recursos de usuário
  - Gerenciar privilégios de usuário
  - **Configurar PAM**

Introdução

Já reparou que, quando cria um usuário, ele já vem populado com arquivos ocultos ou até mesmo diretórios já criados? Isso ocorre porque o diretório `/etc/skel` mantém os esqueletos (templates) dos arquivos.bash_profile e .bashrc que são copiados para o diretório pessoal do usuário no momento que for criada uma conta no sistema; isto é, quando um usuário é criado,

todo o conteúdo desse diretório é copiado para o home do novo usuário.

```
root@kali:/home/kali/707-RHCSA-LFCS/aula_72# ls -la /etc/skel/
total 60
drwxr-xr-x   2 root root  4096 Jul 27  2020 .
drwxr-xr-x 159 root root 12288 Nov 11 10:42 ..
-rw-r--r--   1 root root   220 Feb 25  2020 .bash_logout
-rw-r--r--   1 root root  4261 Jul 27  2020 .bashrc
-rw-r--r--   1 root root  3526 Feb 25  2020 .bashrc.original
-rw-r--r--   1 root root 11759 Jul 20  2020 .face
lrwxrwxrwx   1 root root     5 Jul 20  2020 .face.icon -> .face
-rw-r--r--   1 root root   807 Feb 25  2020 .profile
-rw-r--r--   1 root root  8238 Jul 21  2020 .zshrc
```

Configurar limites de recursos do usuário

Limita o uso de recursos de todo o sistema

```
[root@localhost joatham]# ulimit
```

Os limites podem ser configurados alterando o arquivo `/etc/security/limits.conf`

Configuração típica

1. @student hard nproc 20
2. @faculty soft nproc 20
3. ftp hard nproc 0
4. @student - maxlogins 4

- Os membros do grupo de alunos podem executar apenas 20 processos
- Os membros do grupo de professores receberão informações depois disso, mais de 20 processos foram executados (limite flexível)
- O usuário ftp não pode executar nenhum processo
- Os membros do aluno podem ter no máximo 4 usuários logados. - significa duro e macio

Para manual

```
[root@localhost joatham]# man limits.conf
```

> Os limites serão aplicados na próxima sessão aberta

O comando `ulimit` também pode ser usado para alterar os limites

Configurar PAM

PAM é uma biblioteca que permite autenticar usuários em ambientes como o linux ou unix (Solaris, por exemplo). A necessidade da criação do PAM deveu-se ao problema encontrado quando era preciso fazer o login de um usuário, utilizando uma senha criptografada através de um acesso remoto. Além disso, como cada programa possuía seu método próprio de login, caso fosse necessário mudar o método de autenticação, os programas teriam que ser alterados para reconhecer este novo método.

É neste momento que a Sun Microsystems cria o PAM, um aplicativo centralizador desta tarefa de autenticação, logo não seria necessário cada programa se preocupar com o papel de autenticador, pois esta seria a tarefa do PAM e caso fosse mudado o critério de autenticação, por questões de segurança, somente seria necessário alterar este método no próprio PAM.

A principal vantagem do PAM, além de centralizador das funções de autenticação do login e senha, é que ele é capaz de selecionar, se configurado para tal, os programas aos quais o usuário que fez o login pode ou não acessar. Desta forma, um usuário que quisesse usufruir de aplicativos de áudio e vídeo, por exemplo, remotamente, poderia ser bloqueado o que não aconteceria caso ele estivesse utilizando estes aplicativos localmente.

Linha de Configuração do Módulo PAM

Os arquivos de configuração do PAM, no linux, normalmente estão localizados no diretório `/etc/pam.d/`. Nestes arquivos, a linha de configuração é dada como:

```
service-name module-type control-flag module-path args
```

Divisão dos Módulos

Como o próprio nome diz, o PAM (Plugglable Authentication Modules) é um conjunto de módulos, no qual cada um recebe uma ou mais funções especiais dentro do processo de autenticação. Essa função que cada módulo tem é determinado pelas divisões dos módulos do PAM, que são: AUTH, ACCOUNT, PASSWD, SESSION.

- AUTH - A divisão AUTH trata da autenticação, seja por login/senha ou autenticação

biométrica (voz, retina, impressão digital, por exemplo).

- ACCOUNT - Esta divisão terá o papel de autorização ou não autorização para o uso de programas com base no login, determinando, assim, o usuário apto a utilizar aquele programa ou não.
- PASSWD - A divisão PASSWD é responsável pela troca de senha..
- SESSION - Nesta divisão, determina-se qual será o ambiente do usuário com base no seu login.

Controle de Bandeira

O control-flag é utilizado para indicar de que forma a biblioteca do PAM reagirá ao sucesso ou falha do módulo que está associado. Outra função que a control-flag pode executar é dando prioridades a cada módulo, visto que eles podem ser empilhados.

- REQUIRED - Estabelece que a falha do módulo utilizando a sintaxe REQUIRED, não será mostrada ao usuário até que todos os módulos estejam sendo executados.
- REQUISITE - Parecido com o REQUIRED, porém, no caso de falha, o controle é retornado direto a aplicação. Esta flag é muito utilizada para proteger um usuário que tente colocar sua senha quando o meio está inseguro.
- SUFFICIENT - A falha deste módulo não implica em falha da autenticação como um todo. Se o módulo falhar, o próximo da classe é executado. Se não houver próximo, então a classe retorna com sucesso. Se, por outro lado, o módulo terminar com sucesso, então os módulos seguintes dessa classe não serão executados. Este parâmetro é bastante usado no caso de se usar LDAP para a autenticação, por exemplo, ou outra fonte de dados.
- OPTICIONAL - Módulos marcados como optional praticamente não influenciam o resultado da autenticação como um todo. Eles terão alguma influência somente caso os módulos anteriores da mesma classe não apresentem um resultado definitivo. # Implementar filtragem de pacotes

Pontos de estudo para o exame

Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Rede**
  - Configurar rede e resolução de nome de host estaticamente ou dinamicamente
  - Configurar serviços de rede para iniciar automaticamente na inicialização
  - **Implementar filtragem de pacotes**
  - Iniciar, parar e verificar o status dos serviços de rede
  - Rotear tráfego IP estaticamente

– Sincronizar o tempo usando outros pares de rede

Introdução

- O firewall é gerenciado pelo Kernel
- A funcionalidade do firewall do kernel é o Netfilter
- O Netfilter processará as informações que entrarão e sairão do sistema

Para isso possui duas tabelas de regras chamadas de chains: - INPUT que contém regras aplicadas aos pacotes que entram no sistema - OUTPUT que contém regras aplicadas a pacotes que saem do sistema

Outra cadeia pode ser usada se o sistema estiver configurado como roteador: FORWARD

Finalmente, existem outras duas cadeias: PREROUTING, POSTROUTING

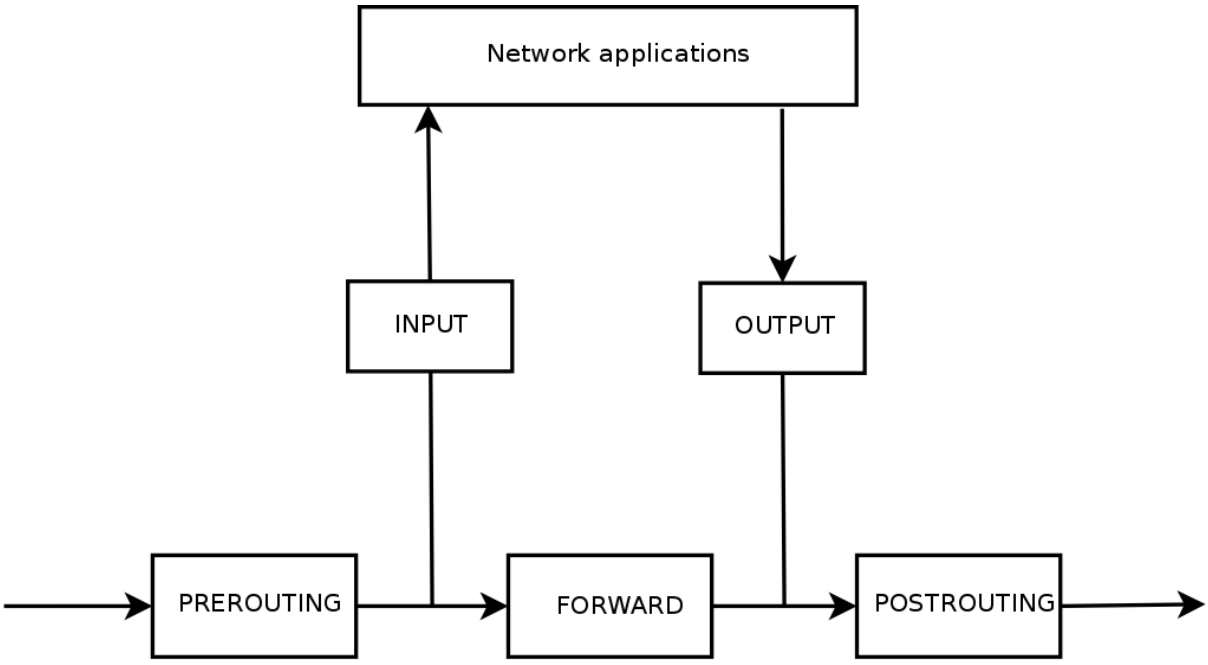


Fig. 32.1: netfilter

A imagem mostra a ordem com que as várias cadeias são avaliadas. As setas indicam a rota dos pacotes:

- Os pacotes de entrada são gerados de fora
- Os pacotes de saída são gerados por um aplicativo ou são pacotes em trânsito

As regras dentro das cadeias são avaliadas de maneira ordenada. - Quando uma regra coincide,

as outras regras são ignoradas - Se nenhuma regra corresponder, a política padrão será aplicada - Política padrão: - ACCEPT: o pacote será aceito e continuará seu caminho pelas cadeias - DROP: o pacote será rejeitado.

O daemon do `firewalld` pode ser substituído pelo daemon do `iptables` (a configuração que estava em vigor até recentemente)

```
[root@rhel8 joatham]# systemctl stop firewalld
```

```
[root@rhel8 joatham]# iptables -L
```

```
- Saída mais detalhada `iptables -L -v`
- Mostra a configuração das cadeias de iptables
- Observe que as políticas são definidas como ACCEPT para cada cadeia. Isso significa que
    nenhum pacote será rejeitado. Isso é igual a ter um firewall desligado.
```

```
[root@rhel8 joatham]# systemctl disable firewalld
```

```
[root@rhel8 joatham]# yum -y install iptables-services
```

```
[root@rhel8 joatham]# systemctl enable iptables
```

> Com esta configuração as regras devem ser inseridas

Defina a política padrão para DROP para a cadeia INPUT

```
[root@rhel8 joatham]# iptables -P INPUT DROP
```

Sintaxe das regras de iptables:

- `iptables {-A|I} chain [-i/o interface][-s/d ipaddres] [-p tcp|upd|icmp [--dport|--sport nn ]] -j [LOG|ACCEPT|DROP|REJECT]`

- {-A|I} chain
  - -A - Anexar como última regra
  - -I - Inserir. Isso requer um número após a cadeia que indica a posição da regra
- [-i/o interface]
  - Ex -i eth0: o pacote é recebido (entrada) na interface eth0
- [-s/d ipaddres]
  - -s - Endereço de Origem. ipaddres podem ser um endereço ou uma sub-rede
  - -d - Endereço de destino. ipaddres podem ser um endereço ou uma sub-rede
- [-p tcp | upd | icmp [–dport | –sport nn. . .]]
  - -p - protocolo
  - –dport - Porto de destino
  - –sport - Porta de origem
- `-j` [LOG|ACCEPT|DROP|REJECTED]
  - ACCEPT - Aceitar o pacote
  - DROP - Silenciosamente rejeitado
  - REJECT - Rejeite o pacote com um pacote de erro ICMP
  - LOG - Pacote de log. A avaliação das regras não está bloqueada.

Por exemplo:

Aceite todo o tráfego de loopback de entrada

```
[root@rhel8 joatham]# iptables -A INPUT -i lo -j ACCEPT
```

Aceite todo o tráfego de loopback de saída

```
[root@rhel8 joatham]# iptables -A OUTPUT -o lo -j ACCEPT
```

Aceite todo o tráfego de entrada para a porta 22 tcp

```
[root@rhel8 joatham]# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Esta é uma regra usada para ACCEPT todo o tráfego gerado como uma resposta de uma conexão de entrada que foi aceita. Por exemplo, se o tráfego de entrada para o servidor web na porta 80 foi aceito, esta regra permite que o tráfego de resposta saia do sistema sem inserir regras específicas na cadeia OUTPUT

```
[root@rhel8 joatham]# iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

> O arquivo /etc/services contém uma lista de portas conhecidas com nomes de serviços. # Configurar um servidor DNS de cache | Manter uma zona DNS ## Pontos de estudo para o exame Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Configuração de serviço**
  - **Configurar um servidor DNS de cache**
  - **Manter uma zona DNS**
  - Configurar aliases de e-mail
  - Configurar servidores e clientes SSH
  - Restringir o acesso ao servidor proxy HTTP
  - Configurar um serviço IMAP e IMAPS
  - Consultar e modificar o comportamento dos serviços do sistema em vários modos operacionais
  - Configurar um servidor
  - HTTP Configurar log do servidor HTTP arquivos
  - Configurar um servidor de banco de dados
  - Restringir o acesso a uma página da web
  - Gerenciar e configurar contêineres
  - Gerenciar e configurar máquinas virtuais

Introdução

Durante os anos 70, Arpanet era uma pequena comunidade de algumas centenas de hosts. Um único arquivo, HOSTS.TXT, continha toda a informação necessária sobre os hosts.

Este arquivo continha nome para endereçar cada host conectado a ARPANET. O arquivo era mantido pela Network Information Center (NIC) e distribuído por um único host, Stanford Research Institute's Network Information Center (SRI-NIC). Os administradores da ARPANET enviavam ao NIC, por e-mail, quaisquer mudanças que tivessem sido efeituadas e periodicamente SRI-NIC era atualizado, assim comoo arquivo HOSTS.TXT.

As mudanças eram compiladas em um novo HOSTS.TXT uma ou duas vezes por semana. Com o crescimento da ARPANET, entretanto, este esquema tornou-se inviável. O tamanho do arquivo HOST.TXT crescia na proporção em que crescia o número de hosts. Além disso, o tráfego gerado com o processo de atualização crescia em proporções ainda maiores uma vez que cada host que era incluído não só significava uma linha a mais no arquivo HOST.TXT,

mas um outro host atualizando a partir do SRI-NIC. Quando a ARPANET passou a usar protocolos TCP/IP, a população da rede "explodiu", passando a existir alguns problemas com o HOST.TXT:

- Nomes que coincidiam: dois hosts do arquivo HOSTS.TXT não podiam ter o mesmo nome. Porém, apesar do NIC poder designar endereços únicos para cada host, ele não tinha nenhuma autoridade sobre os nomes dados aos mesmos. Não havia nada que pudesse evitar que alguém adicionasse um host comum nome conflitante, interrompendo o sistema de algum outro host já existente.
- Consistência: manter a consistência do arquivo com a rede se expandindo àquelas proporções se tornou cada vez mais difícil. Quando o arquivo conseguia conter todos os hosts, algum host trocava de endereço ou um novo host adicionado tinha quebrado a conexão do host que se desejava acessar. Ironicamente, o sucesso da ARPANET tornou o arquivo HOSTS.TXT falho e obsoleto.

Os administradores da ARPANET tentaram outras configurações que resolvessem o problema do HOST.TXT. O objetivo era criar um sistema que resolvesse os problemas em uma tabela única de hosts. O novo sistema deveria: permitir que o administradorlocal tornasse os dados mundialmente disponíveis; descentralizar a administração a fim de resolver o problema do gargalo gerado por um único host, diminuindo o problema do tráfego.

Além disso, o fato da administração ser local faria com que a atualização dos dados se tornasse uma tarefa bem mais simples. O esquema deveria usar nomes em hierarquia para garantir a exclusividade dos nomes. Paul Mockapetris, do USC's Information Science Institute, foi o responsável pela arquitetura do sistema. Em 1984, ele lançou o RFCs 882 e 883, que descreve o "Domain Name System", ou DNS. Esses RFCs foram sucedidos pelos RFCs 1034 e 1035, que possuem as especificações atuais do DNS.

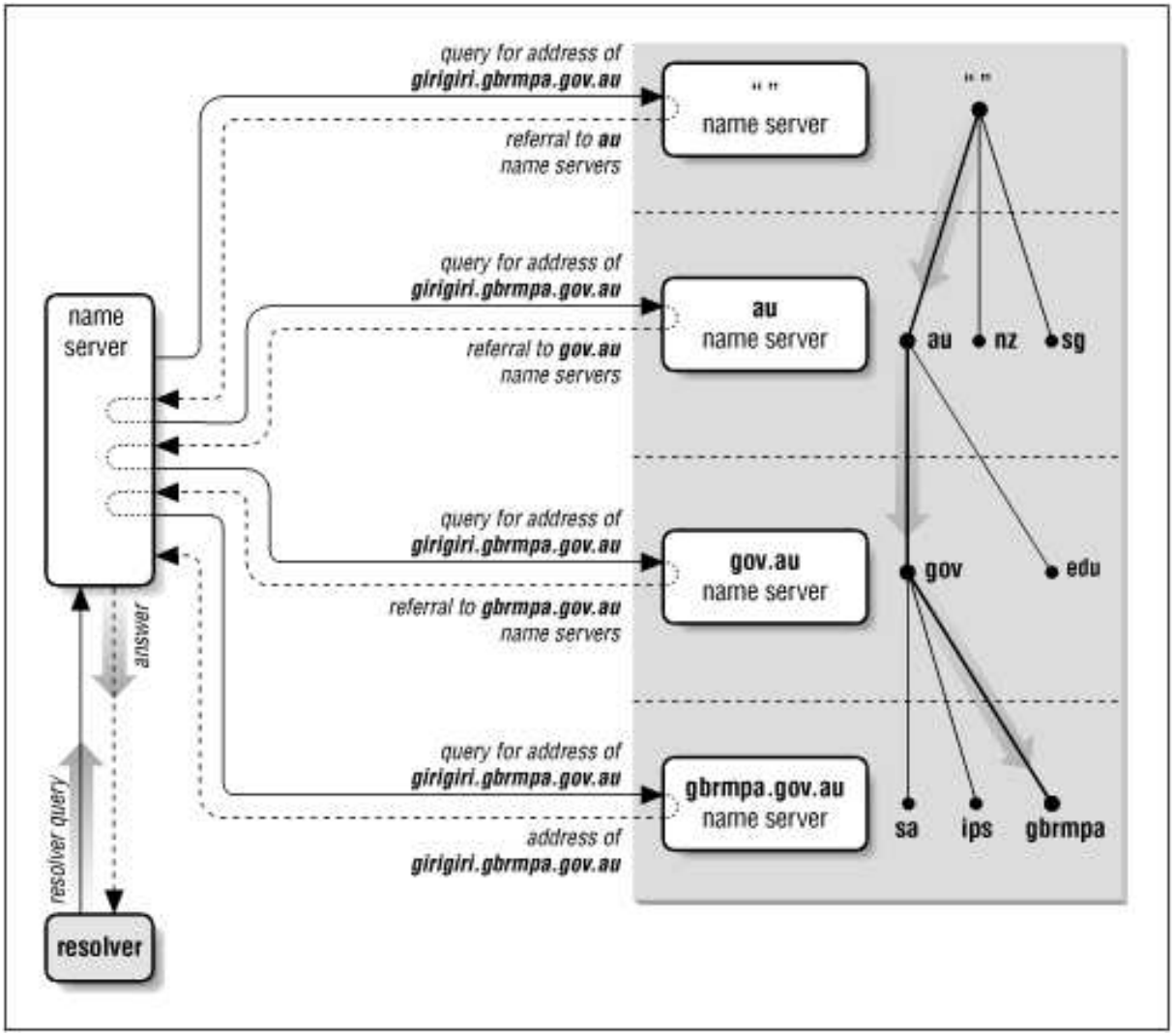


Fig. 32.2: dns

O BIND (Berkeley Internet Name Domain) é o servidor de nomes utilizado na grande maioria dos servidores da Internet, provendo uma estável e robusta arquitetura sobre a qual as organizações podem construir sua estrutura de nomes. Será este servidor que abordaremos no curso. Sua instalação é bem simples:

O servidor DNS Linux está vinculado:

```
[root@rhel8 joatham]# yum -y install bind bind-utils
```

Arquivo de configuração principal `/etc/named.conf`

```
options {
        listen-on port 53 { 127.0.0.1; 192.168.0.0/24; };
        ...
        allow-query        { localhost; 192.168.0.0/24; };
        allow-query-cache  { localhost; 192.168.0.0/24; };
        ...
        recursion yes;
        forwarders {
                8.8.8.8;
                8.8.4.4;
        };
        ...
};

zone "teste.com." IN {
        type master;
        file "/var/named/teste.com.zone";
};

zone "0.168.192.in-addr.arpa" IN {
        type master;
        file "/var/named/rev.teste.com.zone";
};
```

- `listen-on port 53` - Diz em quais interfaces de rede e porta aceitaram as consultas do cliente.
- `allow-query` - Define as redes das quais os clientes podem postar solicitações de DNS.
- `allow-query-cache` - Define os endereços/redes a partir dos quais os clientes têm permissão para fazer consultas que acessam o cache local.
- `forwarders` - Especifica os servidores de nomes para os quais as solicitações de DNS devem ser encaminhadas se não puderem ser resolvidas diretamente.
- `zone` - Contém a configuração do domínio.
  - `file` - Especifica o arquivo onde os dados da zona para o domínio estão localizados.
- `zone "0.168.192.in-addr.arpa"` - é a configuração para zona reversa ou pesquisa reversa. Uma zona reversa permite que o DNS seja convertido de um endereço em um nome.
  - `0.168.192` - Deve ser substituído pelos três primeiros octetos de qualquer intervalo de endereços de rede gerenciado

```
[root@rhel8 joatham]# systemctl start named
```

Manter uma zona DNS

Conteúdo de `/var/named/teste.com.zone`

```
$TTL 3H
```

```
@       IN SOA  dns root.teste.com. (
                                      0       ; serial
                                      1D      ; refresh
                                      1H      ; retry
                                      1W      ; expire
                                      3H )    ; minimum
        IN NS   dns
        IN MX   10 email

dns     IN A    192.168.0.29
email   IN A    192.168.0.29
web     IN A    192.168.0.29
www.web IN CNAME web
```

- Linha 2: Este é o local onde o registro de controle SOA (início de autoridade) começa.
  - `@` - significa que o nome da zona será extraído da entrada correspondente em `/etc/named.conf`(neste exemplo teste.com.)
  - `dns` - é o nome do servidor autorizado para a zona.
  - root.teste.com. - um endereço de e-mail da pessoa responsável por este servidor de nomes. Como o sinal `@` já tem um significado especial, . é inserido aqui. Para `root@test.com` a entrada deve-se ler `root.test.com`.
- Linha 8: o `IN NS` especifica o servidor de nomes responsável por este domínio (servidor autoritativo)
- Linha 9: o registro `MX` especifica o servidor de e-mail que aceita, processa e encaminha e-mails para este domínio
- Últimas linhas: estes são os registros de endereço reais onde um ou mais endereços IP são atribuídos a nomes de host.
  - CNAMES mapeia um nome em outro nome

Conteúdo de /var/named/rev.teste.com.zone

```
$TTL 3H
@       IN SOA  dns.teste.com. root.teste.com. (
                                      0       ; serial
                                      1D      ; refresh
                                      1H      ; retry
                                      1W      ; expire
                                      3H )    ; minimum
        IN NS   dns.teste.com.

29      IN PTR  dns.teste.com.
```

- Linha 2: o arquivo de configuração deve ativar a pesquisa reversa para a rede 192.168.1.0. Dado que a zona é chamada `1.168.192.in-addr.arpa`, não deve ser adicionado aos nomes de host. Portanto, todos os nomes de host são inseridos em sua forma completa - com seu domínio e com um . no final. As entradas restantes correspondem às descritas para

a test.com.zona

- Linha 8: esta linha especifica o servidor de nomes responsável por esta zona. Desta vez, porém, o nome é inserido na sua forma completa com o domínio e um . no final.
- Linha 10: esta é a sugestão de registro de ponteiro nos endereços IP nos respectivos hosts. Apenas a última parte do endereço IP é inserida no início da linha, sem . no final.

Para verificar a resolução do nome é possível usar host:

```
host name_to_resolve dns_server_ip
```

Por exemplo:

```
[root@rhel8 joatham]# host dns localhost
```

Por exemplo, da zona reversa:

```
[root@rhel8 joatham]# host 192.168.0.29 localhost
```

33

Configurar aliases de e-mail

Pontos de estudo para o exame

Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Configuração de serviço**
  - Configurar um servidor DNS de cache
  - Manter uma zona DNS
  - **Configurar aliases de e-mail**
  - Configurar servidores e clientes SSH
  - Restringir o acesso ao servidor proxy HTTP
  - Configurar um serviço IMAP e IMAPS
  - Consultar e modificar o comportamento dos serviços do sistema em vários modos operacionais
  - Configurar um servidor
  - HTTP Configurar log do servidor HTTP arquivos
  - Configurar um servidor de banco de dados
  - Restringir o acesso a uma página da web
  - Gerenciar e configurar contêineres
  - Gerenciar e configurar máquinas virtuais

O que é alias?

Alias quer dizer **pseudônimo**, **apelido** e, em computação, é um comando que permite substituir uma palavra por outras ou por uma cadeia de caracteres. Por isso, é muito usado para abreviação de um comando maior e mais complexo. Nos serviços de e-mail, nada mais é que um endereço alternativo, um e-mail que você nunca abre e recebe as mensagens enviadas para ele encaminhadas para seu e-mail - que usa no dia a dia.

Em outras palavras, um alias de email é um endereço que oculta seu endereço verdadeiro de um destinatário. Os aliases podem ajudar a gerenciar melhor, por exemplo, os emails recebidos e a monitorar fontes de mensagens indesejadas (spam).

Por que você deveria ter um alias?

Além do exemplo acima, é ideal se você tem uma microempresa e é você mesmo quem toma conta de tudo. O alias pode ajudar pessoas com sobrenome complicado, que muita gente escreve errado. Por exemplo: pauloferreira@, pauloferrera@, p.ferreira@, p.ferrera e etc. Os endereços receberão mensagens e enviarão para o e-mail correto.

Também é útil para você enviar os e-mails com o endereço que achar mais adequado. Por exemplo: você abre a caixa de e-mail pessoal, mas responde demandas pelo e-mail profissional ou responde a mensagens de SAC e atendimentos de lojas usando alias.

Para gerenciar spool de correio do seu Linux, instale o seguinte pacote:

```
yum -y install mailx
```

o Comando `mailx` lê o spool de e-mail do usuário

- Envie um e-mail para spool

```
echo "joatham" | mail -s "4linux" root
```

root é o usuário alvo

Para criar um arquivo de edição de alias `/etc/aliases`

Adicionar linha como `root: joatham,root`

Criar um alias para `root` significa que e-mail para o `root` será enviado tambem para joatham e root ao mesmo tempo

Com esta sintaxe será adicionado um endereço de e-mail clássico

```
root: user@test.com
```

> No final das alterações a `/etc/aliases`, execute o comando `newaliases` para aplicar as alterações# Restringir o acesso ao servidor proxy HTTP ## Pontos de estudo para o exame Os candidatos ao exame `LFCS` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Configuração de serviço**
  - Configurar um servidor DNS de cache
  - Manter uma zona DNS
  - Configurar aliases de e-mail
  - Configurar servidores e clientes SSH
  - **Restringir o acesso ao servidor proxy HTTP**
  - Configurar um serviço IMAP e IMAPS
  - Consultar e modificar o comportamento dos serviços do sistema em vários modos operacionais
  - Configurar um servidor
  - HTTP Configurar log do servidor HTTP arquivos
  - Configurar um servidor de banco de dados
  - Restringir o acesso a uma página da web
  - Gerenciar e configurar contêineres
  - Gerenciar e configurar máquinas virtuais

Introdução

Servidor Proxy

Um servidor proxy é um computador dedicado ou um sistema de software executado em um computador que atua como intermediário entre um dispositivo de terminal, como um computador, e outro servidor do qual um usuário ou cliente está solicitando um serviço. O servidor proxy pode existir na mesma máquina que um servidor de firewall ou pode estar em um servidor separado, que encaminha as solicitações por meio do firewall.

Uma vantagem de um servidor proxy é que seu cache pode servir a todos os usuários. Se

um ou mais sites da Internet forem solicitados com frequência, eles provavelmente estarão no cache do proxy, o que aumentará o tempo de resposta do usuário. Um proxy também pode registrar suas interações, o que pode ser útil para solucionar problemas.

Como funcionam os servidores proxy

Quando um servidor proxy recebe uma solicitação de um recurso da Internet (como uma página da Web), ele procura em seu cache local de páginas anteriores. Se encontrar a página, ele a devolve ao usuário sem precisar encaminhar a solicitação à Internet. Se a página não estiver no cache, o servidor proxy, agindo como um cliente em nome do usuário, usa um de seus próprios endereços IP para solicitar a página do servidor na Internet. Quando a página é retornada, o servidor proxy a relaciona à solicitação original e a encaminha ao usuário.

Os servidores proxy são usados para fins legais e ilegais. Na empresa, um servidor proxy é utilizado para facilitar a segurança, o controle administrativo ou os serviços de cache, entre outras finalidades. Em um contexto de computação pessoal, os servidores proxy são usados para permitir a privacidade do usuário e a navegação anônima. Os servidores proxy também podem ser usados para o propósito oposto: monitorar o tráfego e prejudicar a privacidade do usuário.

Para o usuário, o servidor proxy é invisível; todas as solicitações de Internet e respostas retornadas parecem ser diretamente com o servidor de Internet endereçado. (O proxy não é realmente invisível; seu endereço IP deve ser especificado como uma opção de configuração para o navegador ou outro programa de protocolo.)

Os usuários podem acessar proxies da web online ou configurar navegadores da web para usar constantemente um servidor proxy. As configurações do navegador incluem opções manuais e detectadas automaticamente para proxies HTTP, SSL, FTP e SOCKS. Os servidores proxy podem servir a muitos usuários ou apenas um por servidor. Essas opções são chamadas de proxies compartilhados e dedicados, respectivamente. Existem vários motivos para os proxies e, portanto, vários tipos de servidores proxy, geralmente em categorias sobrepostas.

Servidores proxy de encaminhamento e reverso

Os proxies de encaminhamento enviam as solicitações de um cliente para um servidor web. Os usuários acessam proxies de encaminhamento navegando diretamente para um endereço de proxy da web ou definindo suas configurações de Internet. Os proxies de encaminhamento permitem contornar firewalls e aumentam a privacidade e a segurança de um usuário, mas às vezes podem ser usados para baixar materiais ilegais, como materiais protegidos por direitos autorais ou pornografia infantil.

Proxies reversos tratam de forma transparente todas as solicitações de recursos nos servidores de destino sem exigir nenhuma ação por parte do solicitante.

Proxies reversos são usados:

- Para habilitar o acesso indireto quando um site não permite conexões diretas como medida de segurança.
- Para permitir o balanceamento de carga entre os servidores.
- Para transmitir conteúdo interno aos usuários da Internet.
- Para desativar o acesso a um site, por exemplo, quando um ISP ou governo deseja bloquear um site.

Os sites podem ser bloqueados por motivos mais ou menos legítimos. Proxies reversos podem ser usados para impedir o acesso a conteúdo imoral, ilegal ou protegido por direitos autorais. Às vezes, esses motivos são justificáveis e às vezes não. Os proxies reversos às vezes impedem o acesso a sites de notícias onde os usuários podem ver as informações que vazaram.

Eles também podem impedir que os usuários acessem sites onde possam divulgar informações sobre ações do governo ou da indústria. Bloquear o acesso a esses sites pode violar os direitos de liberdade de expressão.

Outros tipos de servidores proxy

Proxies transparentes são normalmente encontrados perto da saída de uma rede corporativa. Esses proxies centralizam o tráfego da rede. Em redes corporativas, um servidor proxy está associado a - ou faz parte de - um servidor de gateway que separa a rede das redes externas (normalmente a Internet) e um firewall que protege a rede de intrusões externas e permite que os dados sejam verificados para fins de segurança antes da entrega a um cliente na rede. Esses proxies ajudam a monitorar e administrar o tráfego de rede, pois os computadores em uma rede corporativa geralmente são dispositivos seguros que não precisam de anonimato para tarefas rotineiras.

- Os proxies anônimos - Ocultam o endereço IP do cliente, usando-os, permitem o acesso a materiais bloqueados por firewalls ou para contornar proibições de endereços IP. Eles podem ser usados para aumentar a privacidade e / ou proteção contra ataques.
- Proxies altamente anônimos - Ocultam até o fato de que estão sendo usados por clientes e apresentam um endereço IP público não proxy. Portanto, eles não apenas ocultam o endereço IP do cliente que os usa, mas também permitem o acesso a sites que podem bloquear servidores proxy. Exemplos de proxies altamente anônimos incluem I2P e TOR .
- Os proxies Socks 4 e 5 - Fornecem serviço de proxy para dados UDP e operações de pesquisa de DNS, além do tráfego da web. Alguns servidores proxy oferecem ambos os

protocolos Socks.
- Os proxies DNS - Encaminham solicitações de serviço de nome de domínio (DNS) de LANs para servidores DNS da Internet enquanto armazenam em cache para maior velocidade.

Hands On

A variável de ambiente `https_proxy` contém o nome do host ou endereço IP do seu servidor proxy. Como qualquer variável de ambiente, as etapas específicas que você usa para defini-la dependem do seu sistema operacional.

Para habilitar o uso de uma variável de ambiente do servidor proxy deve ser configurada a seguinte variavel `http_proxy`

Use um proxy local ouvindo na porta `3128`

```
[root@rhel8 joatham]# export http_proxy=http://127.0.0.1:3128/
```

Usar um proxy remoto no servidor 192.168.0.1, ouvindo na porta 8080 que requer usuário e senha

```
[root@rhel8 joatham]# export http_proxy=http://username:password@192.168.0.1:8080/
```

Desativar o uso de proxy

```
[root@rhel8 joatham]# unset http_proxy
```

> Para a configuração ser de forma permanente insira a variável no arquivo `/etc/environment`# Configurar um serviço IMAP e IMAPS

Pontos de estudo para o exame

Os candidatos ao exame `LFCS` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Configuração de serviço**
  - Configurar um servidor DNS de cache
  - Manter uma zona DNS
  - Configurar aliases de e-mail
  - Configurar servidores e clientes SSH
  - Restringir o acesso ao servidor proxy HTTP
  - **Configurar um serviço IMAP e IMAPS**
  - Consultar e modificar o comportamento dos serviços do sistema em vários modos operacionais
  - Configurar um servidor
  - HTTP Configurar log do servidor HTTP arquivos
  - Configurar um servidor de banco de dados
  - Restringir o acesso a uma página da web
  - Gerenciar e configurar contêineres
  - Gerenciar e configurar máquinas virtuais

O que é protocolo IMAP?

Vamos partir pela definição: IMAP — *Internet Message Access Protocol* — é um meio de acesso entre o servidor de e-mails e o software cliente. Essa comunicação é feita por meio do protocolo TCP/IP.

Por padrão, o protocolo `IMAP` utiliza a porta `143` na transferência simples de mensagens ou `993` em conexões criptografadas via `SSL`. A utilização dessa porta é recomendada para evitar o alto volume de mensagens de spam.

Uma das principais características do protocolo é manter as mensagens no servidor, ou seja, ao acessá-la por meio do software cliente, os e-mails descarregados não são apagados. Além disso, todas as mensagens são controladas por flags. Dessa maneira, há um eficiente controle de status, que permanece o mesmo em todos os dispositivos que acessarem a conta. Assim, elas podem ter os seguintes status:

- seen: mensagem lida;
- answered: e-mail respondido;
- flagged: mensagem marcada para acompanhamento;
- deleted: e-mail marcado para remoção;
- draft: armazenada na pasta de rascunhos;
- recent: mensagem recebida recentemente.

Outra característica do protocolo é permitir a criação de lista branca e negra na pasta de spam. Esse é um recurso muito importante, pois contribui para o aumento da segurança no e-mail,

já que possibilita tanto o bloqueio de remetentes indesejados, quanto a liberação de e-mails que são enviados indevidamente à pasta de spam. Portanto, facilita a criação de políticas de segurança no correio eletrônico.

Qual é a diferença entre IMAP e POP3?

Os dois protocolos possuem algumas semelhanças, já que os dois têm a função de recuperar as mensagens no servidor e disponibilizá-las para leitura no software cliente. No entanto, as semelhanças não vão muito além disso.

POP3

O protocolo POP3 descarrega as mensagens do servidor apenas no primeiro dispositivo que fez a solicitação. Por exemplo, se você acessa o seu e-mail por um computador de mesa e por um notebook, elas serão descarregadas apenas em um deles, o que fizer a solicitação primeiro. Assim, a outra máquina não terá acesso a esses e-mails.

Apesar de possuir o recurso de manter cópias das mensagens no servidor, essa funcionalidade não é recomendada, já que isso não faz com que alterações feitas nesses e-mails sejam refletidas no servidor. Ou seja, depois de descarregar uma mensagem, qualquer ação, como respondê-la, encaminhá-la ou excluí-la, não será atualizada no servidor, já que você estará manipulando-a localmente.

Como no POP3 as mensagens são descarregadas no software cliente, é extremamente importante manter o backup atualizado da máquina. Esse procedimento não pode ser negligenciado, pois caso haja algum problema, não há como recuperar os e-mails perdidos, uma vez que todos são excluídos da origem.

IMAP

Já o IMAP possui recursos mais avançados. O protocolo faz a sincronização das mensagens em qualquer dispositivo que se conecte ao servidor. Desse modo, é possível acessar a caixa postal tanto pelo celular quanto pelo notebook. Ele permite trabalhar com o correio eletrônico em diferentes modos: offline, online e desconectado.

Além disso, inclui operações para criar, deletar e renomear pastas para armazenar mensagens, que também podem ser movidas entre elas. Por permitir o acesso por vários dispositivos, todos os clientes de e-mail acessam o mesmo conteúdo em todas as pastas que foram criadas na conta.

É importante ressaltar que, ao acessar as contas de e-mail por um software cliente, as mensagens são armazenadas em cache na máquina local, o que permite visualizá-las no modo

offline ou desconectado. No entanto, as alterações feitas nesse momento não são refletidas no servidor, ou seja, para excluí-las ou alterá-las, é preciso sempre estar online.

Hands On

O servidor usado para gerenciar o protocolo `IMAP` é `dovecot`

```
[root@rhel8 joatham]# yum -y install dovecot
```

Configuração básica

```
[root@rhel8 joatham]# vi /etc/dovecot/dovecot.conf
```

```
protocols = imap pop3
```

Isso habilitará o protocolo `imap` e `pop3`

```
[root@rhel8 joatham]# vi /etc/dovecot/conf.d/10-mail.conf
```

```
mail_location = maildir:~/Maildir
```

Isso indica para o servidor onde está localizado o arquivo de e-mail

```
[root@rhel8 joatham]#  cat /etc/dovecot/conf.d/10-ssl.conf
```

Nada a mudar, a configuração padrão habilitará a versão SSL dos protocolos que estão habilitados em dovecot.conf

> Teste através do comando mutt# Configurar um servidor de banco de dados ## Pontos de estudo para o exame Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Configuração de serviço**
  - Configurar um servidor DNS de cache
  - Manter uma zona DNS
  - Configurar aliases de e-mail
  - Configurar servidores e clientes SSH
  - Restringir o acesso ao servidor proxy HTTP
  - Configurar um serviço IMAP e IMAPS
  - Consultar e modificar o comportamento dos serviços do sistema em vários modos operacionais
  - Configurar um servidor
  - HTTP Configurar log do servidor HTTP arquivos
  - **Configurar um servidor de banco de dados**
  - Restringir o acesso a uma página da web
  - Gerenciar e configurar contêineres
  - Gerenciar e configurar máquinas virtuais

Introdução

O MariaDB é um dos sistemas de gerenciamento de banco de dados mais usados para sites e servidores. Instalar MySQL é algo que todo desenvolvedor ou dono de site deveria aprender.

Instale o MariaDB

Você pode seguir adiante e instalar MariaDB com o comando:

```
[root@rhel8 joatham]# yum -y install mariadb mariadb-server
```

Uma lista de pacotes será fornecida e você receberá uma confirmação para baixá-los. Digite y (yes) e pressione Enter para cada uma das solicitações.

Inicie o MariaDB e Verifique o Funcionamento

Uma vez que o MariaDB estiver pronto no CentOS 7, ele não será iniciado automaticamente logo após a instalação. Portanto, você precisa iniciá-lo automaticamente com o seguinte comando:

```
[root@rhel8 joatham]# systemctl start mariadb
```

Você receberá uma resposta uma vez que o MariaDB iniciar, então use o comando abaixo para checar se ele está funcionando de modo adequado:

```
[root@rhel8 joatham]# systemctl status mariadb
```

Como Mudar a Senha do Usuário Root no MariaDB

Na hora de instalar o MariaDB no CentOS 7, uma senha root temporária é gerada. Use o comando abaixo para vê-la:

Primeiramente, execute o seguinte comando:

```
[root@rhel8 joatham]# mysql_secure_installation
```

Insira a senha temporária e então a seguinte mensagem vai aparecer:

```
The existing password for the user account root has expired. Please set a new password.

New password:

Re-enter new password:
```

Como Checar a Versão Atual do MySQL

Uma vez que você tiver instalado o CentOS 7, é possível testar se tudo foi configurado corretamente ao checar a sua versão. Digite o seguinte comando:

```
[root@rhel8 joatham]# mysql -u root -p
```

Insira a senha root que você criou e a resposta será parecida com a seguinte:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.20

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names
     may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Como Gerenciar as Permissões MySQL de Usuário

Conceda acesso ao novo usuário para um banco de dados ao digitar:

```
GRANT ALL PRIVILEGES ON novadb.* TO 'nomedeusuário'@'localhost'
```

Você também pode conceder privilégios de maneira individual, incluindo:

- **SELECT** – Usuários pode ler os bancos de dados usando o comando select
- **CREATE** – Eles podem gerar novas tabelas
- **DROP** – Permite que os usuários removam tabelas
- **DELETE** – Usuários podem remover linhas de tabelas
- **INSERT** – Permite que os usuários adicionem linhas nas tabelas
- **UPDATE** – Permite que eles atualizem as linhas
- **GRANT OPTION** – Conceder ou remover os privilégios de outros usuários

Outros Comandos MariaDB Úteis

O MariaDB também possui uma lista de outros comandos úteis. Apenas digite `\h` ou help para ver a lista exibida abaixo:

Lista de todos os comandos MariaDB:

Nota que todos os comandos de texto devem ser os primeiros da linha e terminar com ';'

```
?         (\?) Sinônimo para `ajuda`.
clear     (\c) Comando limpar.
connect   (\r) Reconectar ao servidor. Argumentos opcionais são db e host.
delimiter (\d) Define o delimitador da declaração. NOTA: Faz com que o resto da linah seja
     considerada um novo delimitador.
edit      (\e) Comando de edição com o $EDITOR.
ego       (\G) Enviar comando para o servidor mysql, exibindo resultados verticalmente.
exit      (\q) Sair do msql. Mesmo que quit.
go        (\g) Enviar o comando para o servidor mysql.
help      (\h) Exibe esta ajuda.
nopager   (\n) Desabilita o pager, imprime o stdout.
notee     (\t) Não escreve no outfile.
```

```
pager     (\P) Define o PAGER [to_pager]. Imprime os resultados da solicitação via PAGER.
print     (\p) Imprime o comando atual.
prompt    (\R) Modifica o seu promt do mysql.
quit      (\q) Sair do mysql.
rehash    (\#) Reconstrui o hash de conclusão.
source    (\.) Executa um arquivo de script SQL. Usa um nome de arquivo como um argumento.
status    (\s) Obtém informações de status do servidor.
system    (\!) Executa um comando de shell do sistema.
tee       (\T) Define o outfile [to_outfile]. Anexa tudo no outfile fornecido.
use       (\u) Usa outro banco de dados. Toma o nome do banco de dados como um argumento.
charset   (\C) Mudar para outro charset. Pode ser necessário para processar o binlog com
     charsets multi-byte.
warnings  (\W) Mostra avisos depois de cada argumento.
nowarning (\w) Não mostra avisos depois de cada argumento.

Para ajudar sobre o lado do servidor, digite 'help contents'
mysql>
```

34

Gerenciar e configurar máquinas virtuais

Pontos de estudo para o exame

Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Configuração de serviço**
  - Configurar um servidor DNS de cache
  - Manter uma zona DNS
  - Configurar aliases de e-mail
  - Configurar servidores e clientes SSH
  - Restringir o acesso ao servidor proxy HTTP
  - Configurar um serviço IMAP e IMAPS
  - Consultar e modificar o comportamento dos serviços do sistema em vários modos operacionais
  - Configurar um servidor
  - HTTP Configurar log do servidor HTTP arquivos
  - Configurar um servidor de banco de dados
  - Restringir o acesso a uma página da web
  - Gerenciar e configurar contêineres
  - **Gerenciar e configurar máquinas virtuais**

Introdução

Máquina virtual baseada em kernel (KVM) é um software de virtualização para CentOS ou RHEL 8. `KVM` transforma seu servidor em um `hipervisor`. Certifique-se de que a Virtualization Technology (VT) esteja habilitada no BIOS do seu servidor.

Hands On

Isso instalará todas as ferramentas necessárias para gerenciar e configurar máquinas virtuais.

```
[root@rhel8 joatham]# yum install qemu-kvm qemu-img libvirt virt-install libvirt-client
```

Isso vai iniciar o daemon para gerenciar ambientes virtuais

```
[root@rhel8 joatham]# systemctl start libvirtd
```

Gerenciar volume de armazenamento

- Conceitos:
  - Pool de armazenamento -> Contêiner de volumes de armazenamento (por exemplo, diretório, partições)
  - Volume de armazenamento -> disco virtual
- Crie um pool de armazenamento:
  - virsh pool-define-as spool dir - - - - "/media/vdisk/
  - virsh pool-build
  - virsh pool-start
  - virsh pool-autostart
- Nos arquivos `/etc/libvirt/storage/*.xml`, você pode encontrar informações sobre o pool de armazenamento
- Crie um disco virtual O tamanho será 1G.
  - [root@rhel8 joatham]# qemu-img create -f raw /media/vdisk/disk.img 1G
- Gerenciar máquinas virtuais

Se você sabe que esse root será capaz de executar máquinas virtuais, descomente `user=root` e `group=root` em `/etc/libvirt/qemu.conf` reinicie o daemon `libvirtd` com `systemctl restart libvirtd`

- Crie uma máquina virtual

```
[root@rhel8 joatham]# virt-install --name=rhel7 --disk path=/mnt/personal-data/SPool1/SVol1
     .img,size=2 --vcpu=1 --ram=1024 --location=/run/media/dos/9e6f605a-f502-4e98-826e-
     e6376caea288/rhel-server-7.0-x86_64-dvd.iso --network bridge=virbr0 --graphics none --
     extra-args console=ttyS0
```

```
- Isso preparará uma nova máquina virtual chamada rhel7 com 1 cpu virtual, 1G de RAM e um
     disco virtual de 2G.

- Após a criação, a máquina virtual será inicializada pela primeira vez e uma imagem ISO
     fornecida será executada. Normalmente ISO será um disco de instalação do sistema
     operacional
- A máquina virtual está configurada para não usar ambiente gráfico e mais uma configuração
      para permitir uma conexão da máquina local é definida
```

Gerenciamento de máquina virtual

Liste todas as máquinas virtuais disponíveis em qualquer estado

```
[root@rhel8 joatham]# virsh list --all
```

Inicie uma máquina virtual chamada rhel7

```
[root@rhel8 joatham]# virsh start rhel7
```

Desligue a máquina virtual chamada rhel7

```
[root@rhel8 joatham]# virsh shutdown rhel7
```

Desligamento forçado de uma máquina virtual chamada rhel7

```
[root@rhel8 joatham]# virsh destroy rhel7
```

Exclua uma máquina virtual chamada rhel7

```
[root@rhel8 joatham]# virsh undefine rhel7
```

Estabeleça uma conexão com a máquina virtual chamada rhel7

```
[root@rhel8 joatham]# virsh console rhel7
```

Configure a máquina virtual para reiniciar se a máquina de hospedagem for reiniciada

```
[root@rhel8 joatham]# virsh autostart rhel7
```

Desativar inicialização automática

```
[root@rhel8 joatham]# virsh autostart --disable rhel7
```

Editar máquina virtual

Mostra informações da máquina virtual

```
[root@rhel8 joatham]# virsh dominfo rhel7
```

Edite o arquivo de configuração da máquina virtual chamada rhel7

```
[root@rhel8 joatham]# virsh edit rhel7
```

Mostra o número de cpu virtual

```
[root@rhel8 joatham]# virsh vcpucount rhel7
```

```
- **Configuração máxima**: especifica o número máximo de CPUs virtuais que podem ser
    disponibilizadas para o servidor virtual após a próxima reinicialização.

- **Máximo ao vivo**: especifica o número máximo de CPUs virtuais que podem ser
```

```
     disponibilizadas para o servidor virtual em execução ou pausado. Se você alterar o
     máximo, isso pode ser diferente até que a máquina virtual seja reinicializada

- **Configuração atual**: especifica o número real de CPUs virtuais que estarão disponíveis
     para o servidor virtual na próxima reinicialização.

- **Atual ao vivo**: especifica o número real de CPUs virtuais que estão disponíveis para o
     servidor virtual em execução ou pausado
```

Ele define o número máximo de cpu virtual no arquivo de configuração para 2. Requer a reinicialização da máquina virtual para ser aplicada. Após a reinicialização, o máximo ao vivo será alinhado

```
[root@rhel8 joatham]# virsh setvcpus --count 2 rhel7 --maximum --config
```

Ele define a configuração da máquina virtual. Este valor é o valor com o qual a máquina virtual será inicializada

```
[root@rhel8 joatham]# virsh setvcpus --count 2 rhel7 --config
```

Defina o número de cpu virtual (ao vivo atual). O número deve ser menor ou igual ao máximo ao vivo. Você não pode remover CPUs virtuais de um servidor virtual em execução:

```
[root@rhel8 joatham]# virsh setvcpu --count 2 rhel7
```

Define a quantidade máxima de memória da máquina virtual. A máquina virtual deve estar desligada:

```
[root@rhel8 joatham]# virsh setmaxmem --size 2G rhel7
```

Define a quantidade de memória da máquina virtual. A máquina virtual deve estar em execução:

```
[root@rhel8 joatham]# virsh setmem --size 2G rhel7
```

35

Criar e configurar armazenamento criptografado

Pontos de estudo para o exame

Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Gerenciamento de armazenamento**
  - Listar, criar, excluir e modificar partições de armazenamento físico
  - Gerenciar e configurar armazenamento LVM
  - **Criar e configurar armazenamento criptografado**
  - Configurar sistemas para montar sistemas de arquivos durante ou durante a inicialização
  - Configurar e gerenciar espaço de troca
  - Criar e gerenciar dispositivos RAID
  - Configurar sistemas para montar sistemas de arquivos sob demanda
  - Criar, gerenciar e diagnosticar permissões avançadas do sistema de arquivos
  - Configurar cotas de disco de usuários e grupos para sistemas de arquivos
  - Criar e configurar sistemas de arquivos

Introdução

Segurança e privacidade são dois assuntos muito importantes. Cada um de nós, de uma forma ou de outra, tem dados sigilosos armazenados em seu computador. Embora você possa considerar seus dados bastante seguros em um computador doméstico, em um laptop (ou qualquer dispositivo portátil) a situação é muito diferente. Você carrega seu dispositivo com você e não quer perder todos os seus dados preciosos no caso de serem roubados ou perdidos. É aqui que a criptografia do sistema é útil.

Por que LUKS?

Existem muitos motivos pelos quais as pessoas precisariam criptografar uma partição. Quer eles tenham acesso à privacidade, segurança ou confidencialidade, configurar uma partição criptografada básica em um sistema Linux é bastante fácil. Isso é especialmente verdadeiro ao usar o LUKS, uma vez que sua funcionalidade é construída diretamente no kernel.

Hands On

Para usar o armazenamento criptografado, um módulo do kernel deve ser carregado. Carregue o módulo do kernel `dm_crypt`:

```
[root@rhel8 joatham]# modprobe dm_crypt
```

Para carregar o módulo `dm_crypt` automaticamente quando o sistema for reiniciado, utilize:

```
[root@rhel8 joatham]# echo dm_crypt >> /etc/modules-load.d/dm_crypt.conf
```

Liste todos os módulos do kernel carregados

```
[root@rhel8 joatham]# lsmod
```

Instale o software usado para gerenciar o armazenamento criptografado.

```
[root@rhel8 joatham]# yum -y install cryptsetup
```

Criptografar

Encripta um volume lógico volumename contido em vgname grupo de volumes.

```
[root@rhel8 joatham]# cryptsetup luksFormat /dev/vgname/volumename
```

- Uma senha deve ser fornecida.
- Quando a confirmação for necessária insira um SIM maiúsculo.

> Este comando também pode ser usado com volume físico

Ele abre o volume criptografado e o associa a um novo dispositivo chamado dispositivo_novo. A senha deve ser fornecida:

```
[root@rhel8 joatham]# cryptsetup open --type luks /dev/vgname/volumename dispositivo_novo
```

Ele cria um sistema de arquivos em um novo dispositivo

```
[root@rhel8 joatham]# mkfs.ext4 /dev/mapper/dispositivo_novo
```

> Agora novo, o novo dispositivo pode ser montado

Encerrar dispositivo

Para desmontar o dispositivo use:

```
[root@rhel8 joatham]# cryptsetup close dispositivo_novo
```

Persistência

Adicionar linha abaixo para `/etc/fstab`

```
```

```
/dev/mapper/dispositivo_novo /mnt/mountpoint ext4 defaults 0 0
```

Reinicialize o sistema ou recarregue o gerenciador do sistema

```
systemctl daemon-reload
```

> O novo volume criptografado será montado em /mnt/mountpoint# Criar e gerenciar dispositivos RAID ## Pontos de estudo para o exame Os candidatos ao exame LFCS devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Gerenciamento de armazenamento**
  - Listar, criar, excluir e modificar partições de armazenamento físico
  - Gerenciar e configurar armazenamento LVM
  - Criar e configurar armazenamento criptografado
  - Configurar sistemas para montar sistemas de arquivos durante ou durante a inicialização
  - Configurar e gerenciar espaço de troca
  - **Criar e gerenciar dispositivos RAID**
  - Configurar sistemas para montar sistemas de arquivos sob demanda
  - Criar , gerenciar e diagnosticar permissões avançadas do sistema de arquivos
  - Configurar cotas de disco de usuários e grupos para sistemas de arquivos
  - Criar e configurar sistemas de arquivos

Introdução

RAID, do inglês *Redundant Array of Independent Disks*, significa **Conjunto Redundante de Discos Independentes**. A ideia básica por trás do RAID é combinar diversos discos e de baixo custo em um conjunto, a fim de atingir objetivos de desempenho ou redundância inatingíveis com um disco grande e de custo alto. Este conjunto de discos aparece para o computador como uma única unidade ou disco dearmazenamento lógico. O conceito fundamental do RAID é que os dados podem ser distribuídos ao longo de cada disco do conjunto de maneira consistente. Para fazer isso, primeiramente os dados precisam ser quebrados em pedaços de tamanho consistente (geralmente de 32KB ou 64KB, apesar de poder usar tamanhos diferentes).

Cada pedaço é, então, gravado em um disco rígido no RAID, conforme o nível do RAID usado. Quando os dados forem acessados, o processo é revertido, dando a impressão de que os discos

múltiplos são um disco grande.

Temos como principais vantagens do RAID: - Ganho de desempenho no acesso para leitura ou gravação; - Redundância em caso de falha em um dos discos; - Uso múltiplo de várias unidades de discos; - Facilidade em recuperação de conteúdo perdido; - Impacto reduzido na falha de um disco.

Tipos de RAID

Os principais níveis de RAID utilizados hoje no mercado são os níveis 0, 1, 5, 6 e suas derivações, como, por exemplo, o RAID 10. Vamos entendê-los:

- RAID 0 (Stripping): este é o único nível de RAID que não implementa redundância. Sua finalidade é aumentar o desempenho de leitura e gravação, uma vez que ao gravar, divide os dados em partes iguais e armazena cada fragmento em um disco diferente simultaneamente. Por isto, com dois discos, a velocidade de leitura praticamente dobra. Com três discos, triplica. E assim por diante. São necessários ao menos dois discos para implementar RAID 0 e eles podem ser de tamanhos diferentes.

> Então o RAID 0 garante redundância?

> Não!!! Esta é a desvantagem, pois se qualquer um dos discos falhar, o sistema operacional para de funcionar, além de ocasionar perda dos dados. Portanto, é um método que requer cuidados!

- RAID 1 (Mirroring): O nível mais utilizado. Sua principal finalidade é prover redundância dos dados. Esta é garantida pela duplicação dos dados que são gravados em cada par de discos, logo, se um deles falhar, o outro continuará operando e mantendo a informação disponível, até que a substituição do disco defeituoso seja feita. O ganho de desempenho está na leitura, uma vez que os dados são lidos em partes iguais e simultaneamente de todos os discos. A desvantagem desse nível é que só metade do volume total de armazenamento nos discos utilizados ficará disponível para o sistema operacional. É preciso no mínimo dois discos para implementar RAID1, sempre em pares.

> RAID1 é backup? Não!!!!! É apenas redundância.

- RAID 5: Neste nível de RAID, teremos um balanço das vantagens e desvantagens do níveis anteriores, ou seja, RAID 5 provê um ganho de desempenho e tolerânciaa falhas a

custos menores que `RAID 0` ou `RAID 1` individualmente. O ganho dedesempenho está mais uma vez na leitura. Quanto mais discos forem adicionadosa composição, mais rápida será a leitura, uma vez que a gravação é distribuída em blocos de tamanhos iguais por todos os discos.

Hands On

Instala software para gerenciar dispositivos RAID

```
[root@rhel8 joatham]# yum -y install mdadm
```

- RAID 0 - Distribuído - Sem sobressalente

```
[root@rhel8 joatham]# mdadm --create --verbose /dev/md0 --level=stripe --raid-devices=2 /
     dev/sdb1 /dev/sdc1
```

- RAID 1 - espelho

```
[root@rhel8 joatham]# mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/
     sdb1 /dev/sdc1
```

- RAID 5 - (1 paridade + 1 sobressalente)

```
[root@rhel8 joatham]# mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/
     sdb1 /dev/sdc1 /dev/sdd1 --spare-devices=1 /dev/sde1
```

- RAID 6 - (2 paridade + 1 sobressalente)

```
[root@rhel8 joatham]# mdadm --create --verbose /dev/md0 --level=6 --raid-devices=4 /dev/
     sdb1 /dev/sdc1 /dev/sdd1 /dev/sde --spare-devices=1 /dev/sdf1
```

- RAID 10 - (Stripe + Mirror + 1 sobressalente)

```
[root@rhel8 joatham]# mdadm --create --verbose /dev/md0 --level=10 --raid-devices=4 /dev/sd
     [b-e]1 --spare-devices=1 /dev/sdf1
```

Mostra o status do dispositivo RAID. Para usar o dispositivo `md0`, formate-o e use como um dispositivo clássico.

```
[root@rhel8 joatham]# mdadm --detail /dev/md0
```

Monitorando dispositivos RAID

```
[root@rhel8 joatham]# mdadm --assemble --scan
```

```
[root@rhel8 joatham]# mdadm --detail --scan >> /etc/mdadm.conf
```

```
[root@rhel8 joatham]# echo "MAILADDR root" >> /etc/mdadm.conf
```

```
[root@rhel8 joatham]# systemctl start mdmonitor
```

```
[root@rhel8 joatham]# systemctl enable mdmonitor
```

Adicionar disco

```
[root@rhel8 joatham]# mdadm /dev/md0 --add /dev/sbc2
```

```
[root@rhel8 joatham]# mdadm --grow --raid-devices=4 /dev/md0
```

> Ele adiciona um disco sobressalente e depois que cresce a matriz

Remova o disco

```
[root@rhel8 joatham]# mdadm /dev/md0 --fail /dev/sdc1 --remove /dev/sdc1
```

```
[root@rhel8 joatham]# mdadm --grow /dev/md0 --raid-devices=2
```

> Ele marca o disco como defeituoso e o remove. Depois, o tamanho da matriz deve ser ajustado!

Excluir RAID

```
[root@rhel8 joatham]# mdadm --stop /dev/md0
```

`shell [root@rhel8 joatham]# mdadm --zero-superblock /dev/sbc2`# Configurar cotas de disco de usuários e grupos para sistemas de arquivos ## Pontos de estudo para o exame Os candidatos ao exame `LFCS` devem ser capazes de realizar as tarefas abaixo sem assistência. Agrupamos nas seguintes categorias:

- **Gerenciamento de armazenamento**
  - Listar, criar, excluir e modificar partições de armazenamento físico
  - Gerenciar e configurar armazenamento LVM
  - Criar e configurar armazenamento criptografado
  - Configurar sistemas para montar sistemas de arquivos durante ou durante a inicialização
  - Configurar e gerenciar espaço de troca
  - Criar e gerenciar dispositivos RAID
  - Configurar sistemas para montar sistemas de arquivos sob demanda
  - Criar , gerenciar e diagnosticar permissões avançadas do sistema de arquivos
  - **Configurar cotas de disco de usuários e grupos para sistemas de arquivos**
  - Criar e configurar sistemas de arquivos

Introdução

A utilização de um sistema de quotas é um assunto tão importante quanto dividir o disco rígido em partições. O sistema de quotas serve para limitarmos a quantidadede blocos e "inodes" que um usuário ou grupo pode utilizar em uma determinada partição. Imagine um HD com `100MB` de `home` e 10 usuários.

Se não utilizarmos um sistema de quota por número de blocos é possível que um dos usuários resolva fazer o download de um arquivo de `90MB` utilizando `90%` do espaço disponível, fazendo com que os outros usuários tenham que dividir os outros `10MB` livres.

Se aplicarmosum sistema de quotas, podemos definir que cada usuário utilizará no máximo `10MB`, de forma que cada um terá o mesmo espaço disponível, tornando a divisão justa. Em um cenário como este, resolvemos parte do problema, pois o usuário é capaz decriar um número, suficientemente grande de arquivos com tamanho zero de forma que ele não ocupe os `10MB` atribuídos a ele mas estoure o número máximo de **inodes** que o sistema de arquivos dispõe, impossibilitando, assim, que outro usuário grave qualquer coisa neste sistema de arquivos, mesmo que haja espaço livre.

O sistema de quotas é uma funcionalidade do **filesystem** e do kernel, sendo assim, ambos têm que serem capazes de suportá-lo. Uma vez que o **filesystem** suporta quotas, devemos adicionar os parâmetros de montagem, **usrquota** e **grpquota** ao **filesystem** que utilizaremos com esse sistema. Isso é feito no arquivo `/etc/fstab`. Além disso, temos que criar, na raiz desses **filesystems**, os arquivos de controle, chamados `aquota.user` e `aquota.group`. Uma vez criada essa estrutura, basta editar os arquivos de controle de quotas e distribuir as quantidades de forma apropriada. A quota somente pode ser aplicadapor partições.

Hands On

Instala o software necessário para gerenciar a cota

```
[root@rhel8 joatham]# yum -y install quota
```

As opções de montagem `usrquota` e `grpquota` devem ser inseridas para o sistema de arquivos para o qual habilitar a quota (por exemplo, edição `/etc/fstab`) Depois que as opções forem inseridas, remonte a partição para habilitá-las!

Após a remontagem, execute os `quotacheck -mavug` para verificação dos arquivos que tem que ser criados.

Dois arquivos serão criados:

```
aquota.group
```

```
aquota.user
```

Rode o seguinte comando para iniciar sistema de cotas.

```
[root@rhel8 joatham]# quotaon -a
```

Alternativa: - `quotaon -vu /mnt/ponto_de_montagem` - Ele inicia apenas o usuário de quota para um ponto de montagem específico - `quotaon -vg /mnt/ponto_de_montagem` - Ele inicia apenas o grupo de cotas para um ponto de montagem específico

O seguinte comando serve para mostrae a cota do usuário

```
[root@rhel8 joatham]# quota -vu user
```

A cota é especificada em blocos de 1K de tamanho e em número de inode que é o número de arquivos que podem ser criados:

- Hard Limit: valor máximo permitido
- Soft Limit: um limite que pode ser excedido por um período de carência.
- O período de carência padrão é uma semana(7 dias)

> Quando o período de carência é atingido, o soft limit torna-se e o hard limit

Edite o período de carência. É um valor único para todo o sistema

```
[root@rhel8 joatham]# edquota -t
```

Editar cota do usuário

```
[root@rhel8 joatham]# edquota -u user
```

Em cada coluna pode ser inserido um valor para Soft e Hard Limit para blocos e inodes.

> Normalmente os Soft e Hard Limits são configurados igualmente para evitar confusão.

Por fim, use o comando para mostrar uma visão geral da cota atual para cada usuário:

```
[root@rhel8 joatham]# repquota -aug
```