



www.4LINUX.com.br

**Só na 4Linux você
aprende
MUITO MAIS!**

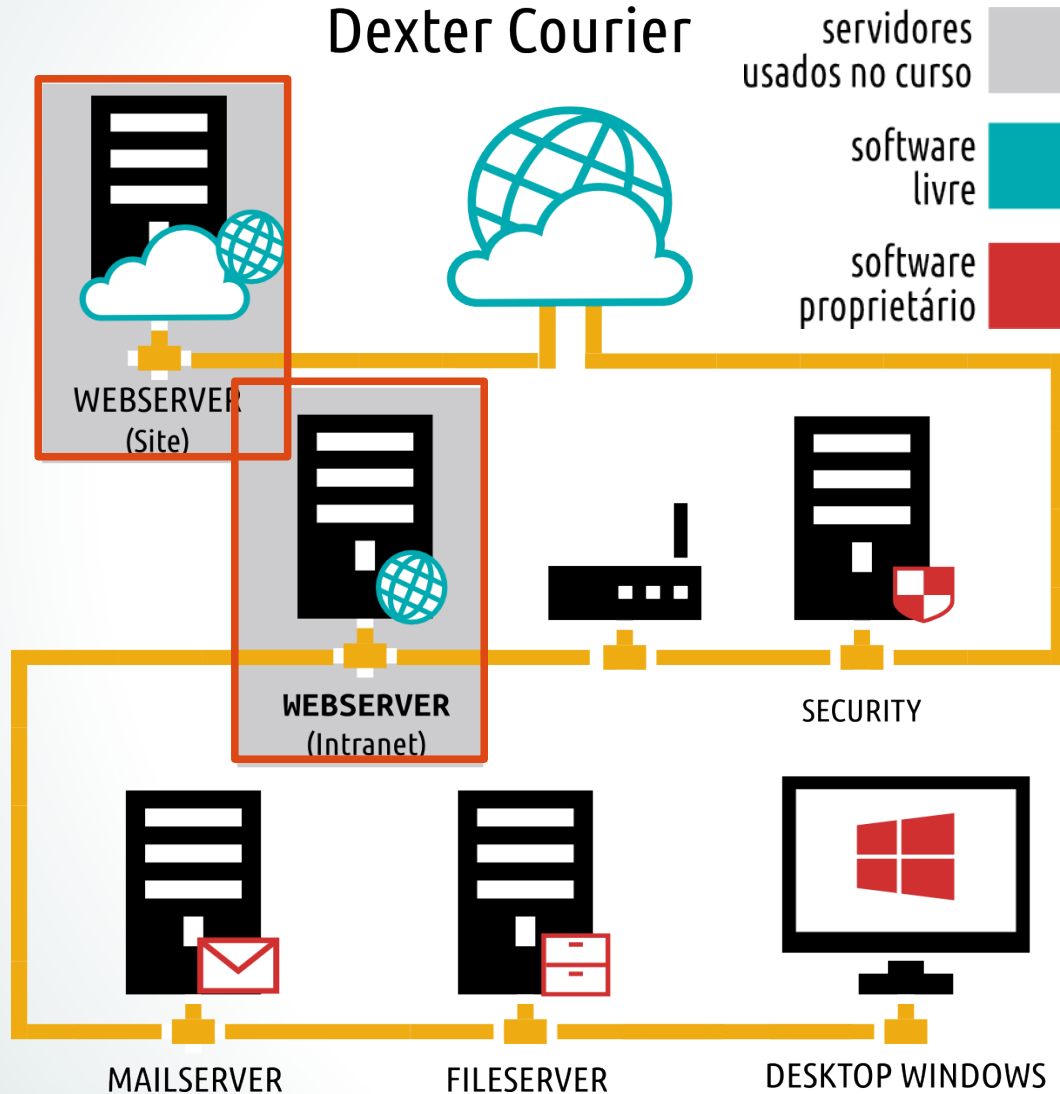
Explorando o Shell do GNU/Linux



IT Experience



Dexter Courier



Nesta Aula:

➤ Usaremos os dois Servidores da Dexter:

➤ WebServerCloud (Site)



➤ WebServerInterno (Intranet)



Objetivos da Aula

➤ Customizar o Servidor para melhor administração:

- O que são Alias;
- Variáveis no Shell;
- Arquivos de Login:
 - /etc/bashrc (~/.bashrc)
 - /etc/profile (~/.bash_profile e .profile)

Explorando o Shell

➤ Identificação de privilégio no Shell:

→ Representa Shell com Privilégio Total (root);

\$ → Representa Shell com Privilégios Limitados (User);

1# whoami

2# su - suporte

3\$ whoami

4\$ exit ou

5\$ logout

6# ↑ ↓ (Histórico)

7# mkd<**TAB**> (Auto Completar)

8# history

9# !<HistoryID>

10# <CTRL+L> (Limpa Tela)

Criando Atalhos de Comandos

➤ Entendendo Alias:

```
1# ls /
```

```
2# ls --color /
```

➤ Criando Alias:

```
3# alias ls='ls --color'
```

```
4# ls /
```

➤ Listando os Alias:

```
5# alias
```

Alias no Shell são como Apelidos de Comandos.

É a forma mais fácil de personalizar comandos do jeito que o administrador precisa.

Um alias pode conter apenas um comando com opções, ou ter um conjunto de comandos.


Laboratório Dexter



- Para facilitar a verificação de usuários existentes no Linux, iremos criar um alias “users” para listar os usuários comuns.

Dica:

 **1#** `awk -F: ' ($3 > 999) { print $1 }' /etc/passwd`

 **1#** `awk -F: ' ($3 > 499) { print $1 }' /etc/passwd`

Laboratório Dexter



- Para facilitar a verificação de usuários existentes no Linux, iremos criar um alias “users” para listar os usuários comuns.

Correção:

```
1# vim /root/.bashrc
alias users="awk -F: ' ( \ $3 > 999) { print \ $1 }' /etc/passwd"

1# vim /root/.bashrc
alias users="awk -F: ' ( \ $3 > 499) { print \ $1 }' /etc/passwd"

2# source /root/.bashrc

3# users
```

Variáveis no Shell

➤ Exibindo o valor de Variáveis:

```
1# echo $HOME ; echo $SHELL
```

➤ Declarando Variável Local:

```
2# VARIAVEL=valor
```

```
3# echo $VARIAVEL
```

```
4# NUM=2
```

```
5# echo $NUM
```

```
6# TEX="Curso Linux"
```

```
7# echo $TEX
```

O Linux usa variáveis para controlar o Sistema.

No shell não é necessário declarar o tipo da variável.

Variáveis como SHELL, HOME são definidas pelo próprio sistema, mas nós também podemos declarar variáveis.

Essa é uma prática usada em Shell Script.

Por nomenclatura, variáveis são definidas com LETRA MAIÚSCULA.

Variáveis no Shell

➤ Declarando Variável Global:

```
1# bash
2# echo $NUM
3# exit
4# export NUM=2
5# bash
6# echo $NUM
7# exit
```

Existem dois Tipos de Variáveis:

LOCAL - Usada apenas no shell que a mesma foi declarada.

GLOBAL - Disponível globalmente em todos os “subshells”

A forma que você declara a variável é que determina se ela é global ou local.

O comando **export** é usado para declarar variáveis Globais.

NOTA: O comando “bash” tem a função de abrir uma nova seção do bash.

Variáveis no Shell

- Listar variáveis Locais e Globais:

```
1# set
```

- Listar variáveis Globais:

```
2# env
```

- Excluindo uma Variável:

```
3# unset NUM
```

```
4# echo $NUM
```

Para definir as variáveis de forma definitiva usamos o arquivo profile.

Cada usuário possui em seu home o arquivo .bash_profile ou .profile

Caso queira definir de forma global para todos os usuários a variável deve ser declarada no /etc/profile.

Variáveis no Shell

➤ Variáveis de Sistema:


- HOME → Define o diretório home do usuário logado;
- PATH → Define os diretórios usados para encontrar os comandos;
- SHELL → Define o shell que está sendo utilizado;
- PWD → Define em qual diretório você está no momento;
- USER → Define o usuário que está logado.

Criando Atalhos de Comandos



➤ Personalizando o shell de forma definitiva:

```
1# vim /root/.bashrc
```

Descomentar as Linhas abaixo:



```
[ export LS_OPTIONS='--color=auto'  
alias ls='ls $LS_OPTIONS'  
alias ll='ls $LS_OPTIONS -l'  
alias l='ls $LS_OPTIONS -lA'  
alias rm='rm -i'  
alias cp='cp -i'  
alias mv='mv -i'
```



```
2# source /root/.bashrc
```

```
3# alias
```

O arquivo `.bashrc` é o responsável por armazenar os alias dos usuários do Sistema.

Cada usuário possui em seu diretório home um arquivo `.bashrc`

É possível também definir alias de forma global, ou seja, que estarão disponíveis para todos os usuários, no arquivo do bash no `/etc/`:

 **`/etc/bash.bashrc`** **`/etc/bashrc`**

Auditando o Shell



➤ Definir a variável TMOUT no SHELL:

```
1# echo $TMOUT
2# export TMOUT=300      (Segundos)
3# set | grep TMOUT
4# vim /etc/profile
   export TMOUT=120
5# source /etc/profile
6# echo $TMOUT
```

A variável TMOUT é responsável por controlar o tempo de inatividade do Shell.

Declarando ela no arquivo /etc/profile qualquer usuário que logar no modo texto e ficar inativo por 300 segundos (5 minutos) será deslogado automaticamente.

Auditando o Shell



➤ Implementar Registro de Data e Hora no History:

```
1# history
2# echo $HISTSIZE
3# echo $HISTFILE
4# vim /root/.bashrc
export HISTFILE=~/.bash_history'
export HISTSIZE=1000
export HISTCONTROL='erasedups'
export HISTTIMEFORMAT='%d-%m-%Y %H:%M- '
5# source /root/.bashrc
6# history
```

A variável HISTFILE define onde será gravado o history do usuário logado.

Para remover os comandos duplicados usa-se a variável HISTCONTROL.

A quantidade de comandos que será armazenado no history é definido na variável HISTSIZE.

A personalização é feita na variável HISTTIMEFORMAT.

History no Cloud



Atenção

No Servidor Cloud, especificamente no Laboratório da 4Linux, o history está desativado, pois o aluno não faz acesso via SSH ou Login e sim por um comando específico do OpenVZ (vzctl). Portanto, as configurações do history só funcionarão durante o tempo em que estiver logado.

BASH Completion

- Auto completar para gerenciamento de pacotes:



```
1# apt-get inst<TAB>  
Humm... Não completou!
```



```
1# yum inst<TAB>  
Humm... Não completou!
```

O auto completar facilita e muito na administração de servidores. Porém esse recurso não é nativo.

É necessário a instalação do pacote **bash-completion** para utilizar essa funcionalidade.

BASH Completion



- Instalar o pacote bash-completion para agilidade no shell;

```
1# apt-get install bash-completion
```

```
2# vim /etc/bash.bashrc
```

```
31 # enable bash completion in interactive shells
```

```
32 if ! shopt -oq posix; then
```

```
33     if [ -f /usr/share/bash-completion/bash_completion ]; then
```

```
34         . /usr/share/bash-completion/bash_completion
```

```
35     elif [ -f /etc/bash_completion ]; then
```

```
36         . /etc/bash_completion
```

```
37     fi
```

```
38 fi
```

```
3# source /etc/bash.bashrc
```

BASH Completion



- Instalar o pacote bash-completion para agilidade no shell;

```
1# yum install bash-completion
```

```
2# logout
```

```
3# yum inst<TAB>
```

Pergunta LPI



1) Qual destes comandos permite que você declare a variável `LD_LIBRARY_PATH` como uma variável global, para que as bibliotecas compartilhadas que estão em `/usr/local/lib` sejam válidas?

- A. `set LD_LIBRARY_PATH = /usr/local/lib`
- B. `export LD_PRELOAD = /usr/local/lib`
- C. `export LD_LIBRARY_PATH = /usr/local/lib`
- D. `env LD_LIBRARY_PATH = /usr/local/lib`

Pergunta LPI



1) Qual destes comandos permite que você declare a variável `LD_LIBRARY_PATH` como uma variável global, para que as bibliotecas compartilhadas que estão em `/usr/local/lib` sejam válidas?

- A. `set LD_LIBRARY_PATH = /usr/local/lib`
- B. `export LD_PRELOAD = /usr/local/lib`
- C. `export LD_LIBRARY_PATH = /usr/local/lib`
- D. `env LD_LIBRARY_PATH = /usr/local/lib`

Resposta: C

Pergunta LPI



2) Qual variável de sistema define a localização dos comandos no Linux?

- A. SHELL
- B. COMMANDS
- C. PWD
- D. Nenhuma das alternativas

Pergunta LPI



2) Qual variável de sistema define a localização dos comandos no Linux?

- A. SHELL
- B. COMMANDS
- C. PWD
- D. Nenhuma das alternativas

Resposta: D



www.4LINUX.com.br