



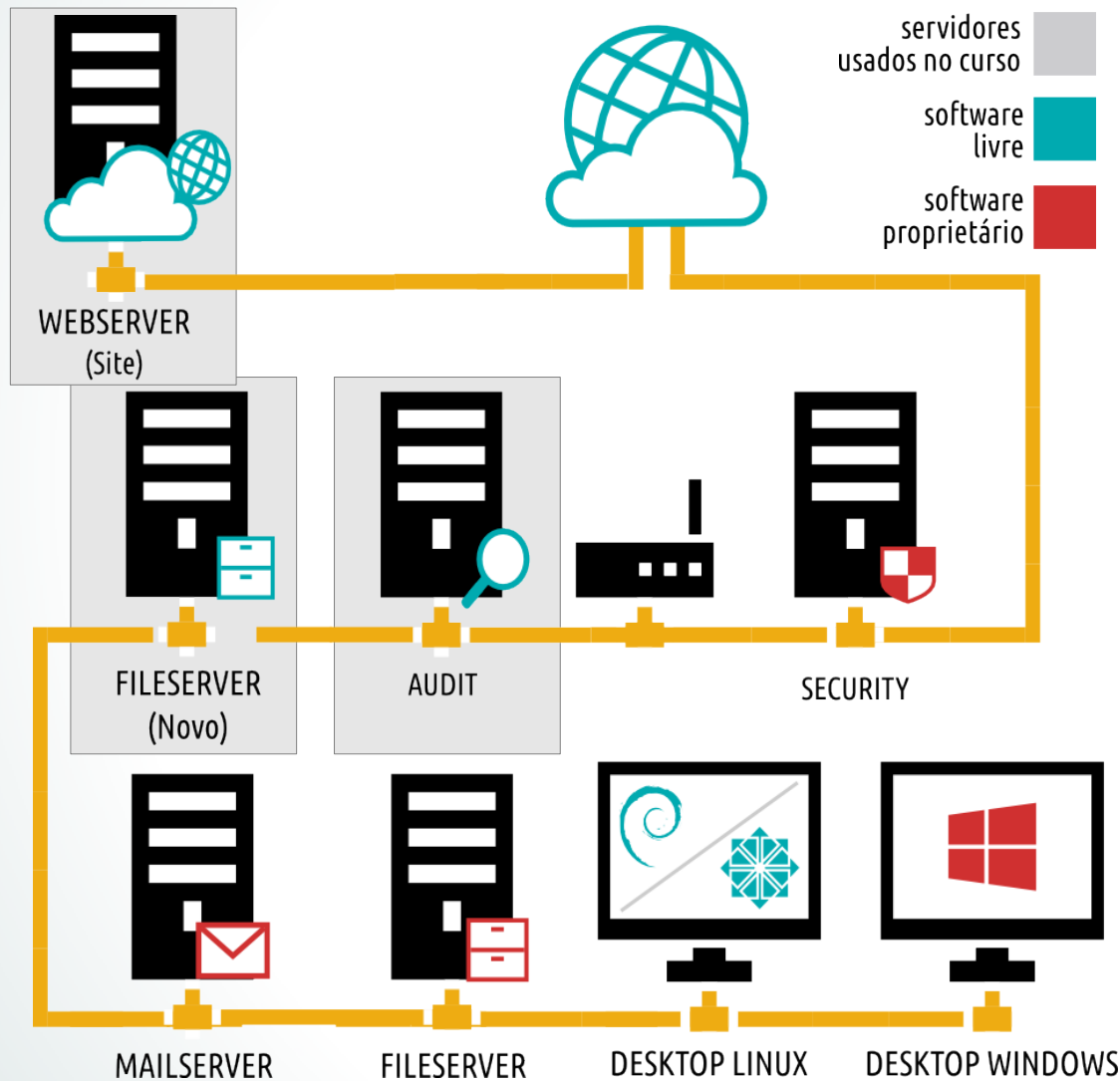
www.4LINUX.com.br

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

Agendamento de Tarefas



IT Experience



Nesta Aula:

- WebServer (Site) - Cloud
Acesso via SSH
SO: Debian GNU/Linux
- WebServer (Intranet) – Local
Acesso pelo VirtualBox
SO: CentOS Linux



Objetivos da Aula

- Entender como funciona o agendamento no sistema;
- Aprender o funcionamento do at;
- Aprender o funcionamento do cron;
- Automatizar o Backup dos Servidores WebServerCloud e WebServerInterno.

Agendamento de Tarefas

➤ Existem dois tipos de ferramentas utilizadas no processo de agendamento de tarefas:

CRON → Agendador de Tarefas com intervalos regulares (Rotinas);

AT → Agendador de Tarefa com Data e Hora Marcada;

➤ Ambos são serviços no Linux e precisam estar iniciados para funcionarem:

```
1# dpkg -l | egrep "at|cron"
```

```
2# rpm -qa | egrep "at|cron"
```

Agendamento de Tarefas

- O **cron** é utilizado para agendar comandos que serão executados periodicamente, ao contrário do **at** que executa comandos pontualmente;
- A “**crontab**” é a tabela através da qual os agendamentos do cron são gerenciados;
- O **daemon crond** verifica essa tabela a cada minuto a procura de agendamentos que devem ser executados;
- Há dois tipos de crontab a de usuarios e a de sistema.

Agendamento de Tarefas

➤ A crontab do sistema é encontrada no arquivo **/etc/crontab** e já possui agendamentos para realizar as tarefas que se encontram nos diretórios **/etc/cron.[hourly|daily|weekly|monthly]**;

```
1# ls /etc/ | grep cron.*
```

```
2# vim /etc/crontab
```

/etc/cron.hourly → Scripts neste diretório serão executado toda hora;

/etc/cron.daily → Scripts neste diretório serão executado todos os dias;

/etc/cron.weekly → Scripts neste diretório serão executado toda semana;

/etc/cron.monthly → Scripts neste diretório serão executado todos os meses.

Agendamento de Tarefas

Crontab – Sintaxe de agendamento e valores válidos

crontab (usuários): minuto hora dia mês dia-da-semana comando;

crontab (sistema): minuto hora dia mês dia da semana USUÁRIO comando;

- **minuto** varia de 0-59;
- **hora** varia de 0-23;
- **dia** varia de 1-31;
- **mes** varia de 1-12;
- **dia-da-semana** varia de 0-7, sendo 0 e 7 são o domingo;
- **usuário** um usuário válido no sistema;
- **comando** o path completo para o comando.

Agendamento de Tarefas

Crontab – Operadores e Segurança

- **Vírgula (,)** → Especifica uma lista de valores, por exemplo: "1,3,4,7,8";
- **Hifen (-)** → Especifica um intervalo de valores, por exemplo: 1-15 (de 1 a 15);
- **Asterisco (*)** → Especifica todos os valores possíveis;
- **Barra (/)** → Especifica “pulos” de valores, por exemplo: se no campo hora utilizarmos "*/4" o comando será executado as "0,4,8,12,16,20";

Agendamento de Tarefas

➤ Por padrão a execução de tarefas pelo cron não trás retorno em tela, dessa forma vamos criar uma tarefa simples e monitorar o log do cron para validar nossos agendamentos:

1 – Abra um arquivo no vim com o seguinte conteúdo:

```
1# vim tarefa.sh
```

```
#!/bin/bash  
echo "Esta é uma tarefa agendada"  
date -s %d+%m+%y
```

O comando **tailf** é utilizado para ver o final, “calda” de um arquivo em tempo real.

2 – Conceda permissão de execução a tarefa:

```
1# chmod +x tarefa.sh
```

3 - Abra um terminal execute o seguinte comando:

```
1# tailf /var/log/cron
```

Agendamento por usuários

➤ A crontab dos usuários pode ser acessada pelo comando **crontab**:

1 – Copie o arquivo tarefa.sh para a home do usuário suporte:

```
1# cp /root/tarefa.sh /home/suporte
```

2 – Verifique a crontab do usuário suporte:

```
2# crontab -lu suporte
```

no crontab for suporte

3 – Vamos criar a crontab para suporte:

```
3# su - suporte
```

```
4# crontab -e
```

Opções do crontab:

- e → Editar o agendamento;
- r → Remove o agendamento;
- l → Lista agendamentos de um usuário;
- u → Especifica o nome do usuário a ser gerenciado.

Agendamento por usuários

4 – Adicione a seguinte tarefa:

```
* /2 * * * * tarefa.sh
```

5 – Salve a tarefa e verifique novamente a crontab do usuário:

```
3# exit
```

```
4# crontab -lu suporte
```

```
5# tailf /var/log/cron.log
```

7 – Apague a crontab do usuário

```
6# crontab -ru suporte
```

Agendamento no Sistema

At – Agendamento único

O comando at é utilizado para criar agendamento únicos no sistema.

Sintaxe: **at HH:mm MM/DD/YYYY**

Vamos agendar uma tarefa de backup:

1# at HH:mm MM/DD/YYYY ← Agende a tarefa para que seja executada em 3 min.

2# at> tar zvzf /backup/etc_bkp.tar.bz2 /etc/

3# at> (Ctrl + d)

4# atq

Nota: O agendamento fica armazenado em arquivos com o nome do usuário dono da tabela no diretório /var/spool/cron/crontabs

Laboratório Dexter



- Iremos agora criar uma rotina com o CRON para automatizar os backups dos Servidores WebServerCloud e WebServerInterno;
- Para o exemplo de sala de aula, iremos criar uma rotina de minutos para ter volume no /backup, num servidor em produção uma rotina diária é o suficiente;
- Começaremos pelo WebServerCloud:

```
1# mkdir /etc/cron.minute/
```

```
2# cd /etc/cron.minute
```

Laboratório Dexter



```
1# vim /etc/cron.minute/backup
```

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

```
tar -cvjf /tmp/bkp_etc_$(hostname -s)_$(date +%d-%m-%y-%M).tar.bz2 /etc/
```

```
tar -cvjf /tmp/bkp_www_$(hostname -s)_$(date +%d-%m-%y-%M).tar.bz2 /var/www/
```

```
2# chmod +x /etc/cron.minute/backup
```

```
3# vim /etc/crontab
```

```
*/15 * * * * root /etc/cron.minute/backup
```

Faça o mesmo processo no Servidor WebServerInterno

Restringindo o uso do Crontab



- É possível restringir o acesso ao cron através dos arquivos “/etc/cron.allow” e “/etc/cron/deny”.
- De forma similar podemos restringir o acesso ao at editando os arquivos “/etc/at.allow” e “/etc/at.deny”.
- Esses controles são lidos cada vez que o usuário tenta adicionar ou apagar uma tabela do cron.

Restringindo o uso do Crontab



➤ Restringindo o acesso do usuário suporte ao cron:

```
1# vim /etc/cron.deny
```

< Adicione uma linha como o nome suporte >

suporte

< Saia do arquivo e tente executar um agendamento >

```
1# su - suporte
```

```
2# crontab -e
```

< Repita o teste utilizando os
agendamentos do at e o arquivo
at.deny >

Caso o mesmo usuário esteja listado nos arquivos cron.allow e cron.deny o usuário será LIBERADO pois o arquivo cron.allow possui prioridade sobre o cron.deny.

O usuário root pode usar o cron sempre, independente dos nomes de usuário listado nos arquivos de controle de acesso.

Pergunta LPI



Para prevenir um específico usuário de usar o agendador de tarefas at, um administrador deverá fazer:

- A. Adicionar o usuário no arquivo `/etc/at.allow`
- B. Adicionar o usuário no grupo `nojobs`
- C. Executar o comando: `atd deny [user]`
- D. Adicionar o usuário na seção `[deny]` do arquivo `atd.conf`
- E. Adicionar o usuário no arquivo `/etc/at.deny`

Pergunta LPI



Para prevenir um específico usuário de usar o agendador de tarefas at, um administrador deverá fazer:

- A. Adicionar o usuário no arquivo /etc/at.allow
- B. Adicionar o usuário no grupo nojobs
- C. Executar o comando: atd deny [user]
- D. Adicionar o usuário na seção [deny] do arquivo atd.conf
- E. Adicionar o usuário no arquivo /etc/at.deny

Resposta: E



www.4LINUX.com.br