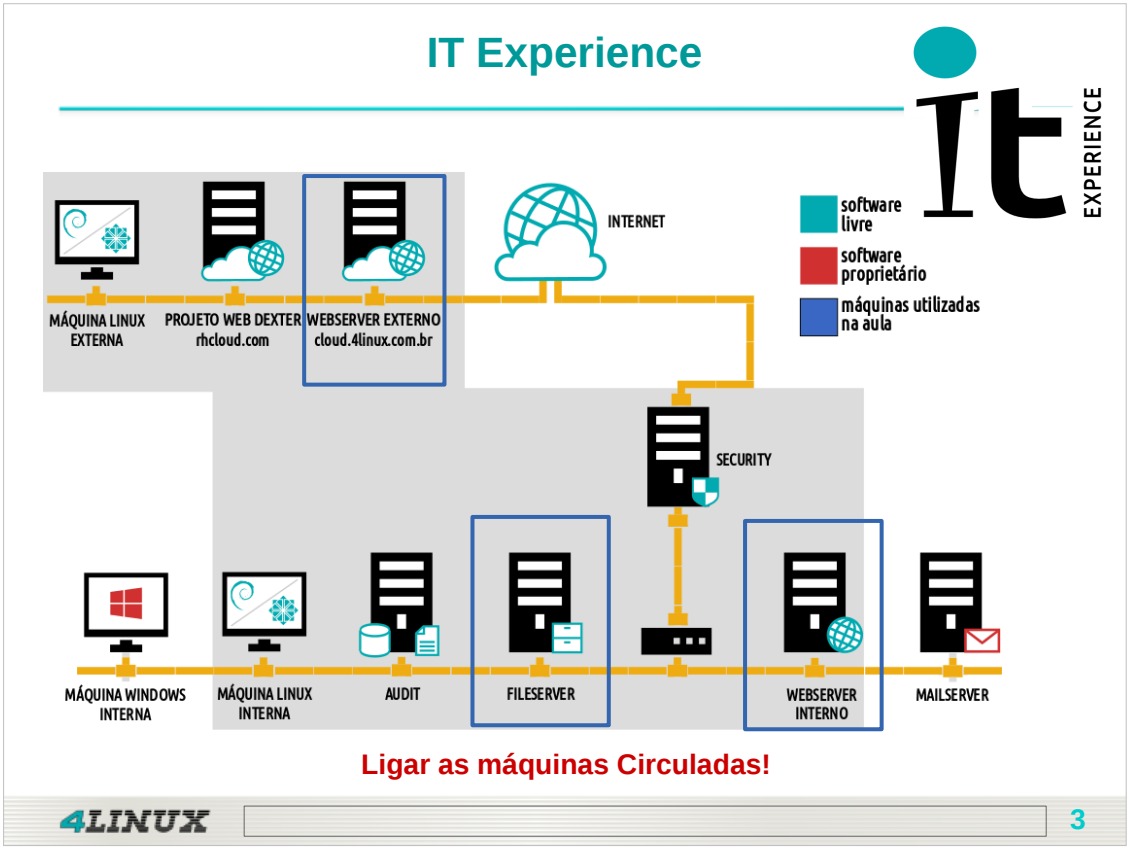
Curso 4451

Linux Security Administration in Cloud

Backup com ferramentas XFS e Agendamento de Tarefas

Anotações:



Anotações:

Objetivos da Aula

Aula 04

- Introdução ao sistema de arquivos XFS;
- Backup completo e incremental com a ferramenta xfsdump;
- Restore completo e incremental com a ferramenta xfsrestore;
- Criar e restaurar backup completo e incremental em rede;
- 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.

Anotações:

Backup com ferramentas XFS

Anotações:

Backup com ferramentas XFS

Sistema de arquivos XFS:

- O XFS é um sistema de arquivos de alta performance com suporte a **journaling**, que teve origem na plataforma IRIX da SGI.
- É completamente multi-processo e pode suportar **grandes sistemas de arquivos** com atributos estendidos, e tamanho de blocos variável.
- O XFS é baseado em extents e utiliza bem o Btrees (diretórios, extensões e espaço livre) para ajudar no ganho de **performance** e **escalabilidade**.

Anotações:

Backup com ferramentas XFS

Servidor: File Server

Criar partições com FDISK:

Para gerenciar novas partições através do FDISK, adicione um novo disco no VirtualBox de 10GB e utilize o seguinte comando:

```
# fdisk /dev/sdc
```


Anotações:

Backup com ferramentas XFS

Servidor: File Server

Criar partições com FDISK:

Dicas de uso:

- Para obter ajuda pressione a tecla “**m**”;
- Para criar uma nova partição pressione a tecla “**n**”;
- Escolha o tipo de partição, sendo “**l**” para lógica e “**p**” para primária;
- Escolha o numero da partição, sendo “**1-4**” para primária e apartir de **5** para lógica;
- Escolha início da partição, sendo “**1**” para o primeiro cilindro;
- Escolha o tamanho da partição, sendo Last cilindro, +cilindros or +sizeK,M,G: **(Enter)**;
- Grave a tabela de partições pressione a tecla “**w**”.

Anotações:

Backup com ferramentas XFS

Servidor: File Server

Pré-requisito para as partições: Sistema de arquivos XFS

1 – Aplique o sistema de arquivos na partição do novo disco:

```
# mkfs.xfs -f /dev/sdc1
```

2 – Crie o ponto de montagem antes de montar:

```
# mkdir /backups-xfs
```

3 – Cfaça a montagem do novo disco e verifique o espaço disponível:

```
# mount /dev/sdc1 /backups-xfs
# df -Th
```


Anotações:

Backup com ferramentas XFS

Servidor: File Server

Pré-requisito para as partições: Sistema de arquivos XFS

4 – Configure a montagem automática no boot para o novo disco:

```
# vim /etc/fstab
….
/dev/sdc1     /backups-xfs     xfs      defaults0  2
```


Anotações:

Backup com ferramentas XFS

Servidor: File Server

Ferramenta xfsdump:

O comando **xfsdump** é utilizado para fazer backup de arquivos com seus atributos em um sistema de arquivos.

Ele examina os arquivos, determina quais precisam ser salvos (backup) e copia esses arquivos para um disco especificado – como fita magnética ou outra mídia de armazenamento.

Anotações:

Backup com ferramentas XFS

Servidor: File Server

Ferramenta xfsdump:

Instalação da ferramenta:

```
# yum install xfsdump
```

Exemplo de linha de comando para backup completo:

```
#  xfsdump -l 0 -p 30 -f  /backups-xfs/backup_boot.0.dump
/boot
# xfsdump -l 0 -p 30 -f /backups-xfs/backup_boot_$(date +
%d-%m-%Y).0.dump  /boot
```


Anotações:

Backup com ferramentas XFS

Servidor: File Server

Ferramenta xfsdump: Opções utilizadas

- **-l**: Especifica um nível de dump de 0 a 9. O nível de dump determina o que sera armazenados no backup;
- **-p**: Faz com que relatórios de progresso sejam impressos no intervalo especificado (intervalo é dada em segundos);
- **-f**: Especifica um destino de despejo. Um destino pode ser o caminho de um dispositivo (como uma unidade de fita), um arquivo regular ou uma unidade de fita remota.

Anotações:

Backup com ferramentas XFS

Servidor: File Server

Incrementando novas informações:

1 – Adicione um novo conteúdo no /media/home:

```
# echo "Novo conteúdo" > /boot/arquivo-novo
```

2 – Crie um backup incremental com a ferramenta xfsdump:

```
#  xfsdump -l 1 -p 30 -f  /backups-xfs/backup_boot.1.dump
/boot
```


Anotações:

Backup com ferramentas XFS

Servidor: WebServer Interno

Ferramenta xfsrestore:

O comando xfsrestore executa a função reversa do xfsdump, podendo restaurar uma cópia de segurança completa de um sistema de arquivos.

Exemplo de linha de comando para restore do backup completo:

```
#  xfsrestore -f  /backups-xfs/backup_boot.0.dump  /tmp
```

Exemplo de linha de comando para restore do backup incremental:

```
#  xfsrestore -f  /media/backup/backup_boot.1.dump  /tmp
```


Anotações:

Backup com ferramentas XFS

Servidor: File Server e WebServer Interno

Backup e restore em rede:

Antes de iniciar o Backup e Restore via rede, torne o usuário "suporte" dono da pasta /backup no servidor WebServer Interno:

```
# chown suporte /backup -R
```

Anotações:

Backup com ferramentas XFS

Servidor: File Server

Backup e restore completo em rede:

Comando para backup completo do servidor File Server para WebServer Interno via SSH:

```
# xfsdump -l 0 -L backup -  /boot | gzip | ssh -p 2222 suporte@192.168.200.20 dd of=/backup/backup_boot_$(date +%d-%m-%Y).gz
```

Comando para restore completo do servidor WebServer Interno para File Server via SSH:

```
# ssh -p 2222 suporte@192.168.200.20 "dd if=/backup/backup_boot_05-08-2015.gz" | gunzip -c | xfsrestore - /tmp
```


Anotações:

Backup com ferramentas XFS

Servidor: File Server

Backup e restore incremental em rede:

1 – Adicione um novo conteúdo no /media/home:

```
# echo "Backup em Rede" > /boot/backup-em-rede
```


Anotações:

Backup com ferramentas XFS

Servidor: File Server

Backup e restore incremental em rede:

Comando para backup incremental do servidor File Server para WebServer Interno via SSH:

```
# xfsdump -l 1 -L backup -   /boot | gzip | ssh -p 2222
suporte@192.168.200.20                                    dd
of=/backup/backup_boot_incremental_01_$(date +%d-%m-%Y).gz
```

Comando para restore incremental do servidor WebServer Interno para File Server via SSH:

```
#    ssh    -p    2222    suporte@192.168.200.20        "dd
if=/backup/backup_boot_incremental_05_08-03-2015.gz"    |
gunzip -c | xfsrestore - /tmp
```


Anotações:

Fundamentação

A “crontab” é utilizada para agendar comandos que serão executados periodicamente, ao contrário do comando “at”, que executa comandos pontualmente.

Há dois tipos de “crontab”: a de usuários e a do sistema. Ambas são arquivos que contem tabelas com informação de quando o comando especificado deve ser executado, sendo que cada linha corresponde a um único agendamento.

A “crontab” é gerenciada pelo “daemon crond”, que a cada um minuto verifica se há algum agendamento que deve ser executado e, se houver, executa-o.

Agendamento de Tarefas

Servidor: WebServer Interno

- Existem dois tipos de ferramentas utilizadas no processo de agendamento de tarefas:
  - **CRON** → Agendador de tarefas com intervalos regulares (rotinas);
  - **AT** → Agendador de tarefas 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"
```


Anotações:

Agendamento de Tarefas

Servidor: WebServer Interno

- 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, à procura de agendamentos que devem ser executados;
- Há dois tipos de crontab: a de usuários e a de sistema.

Anotações:

Agendamento de Tarefas

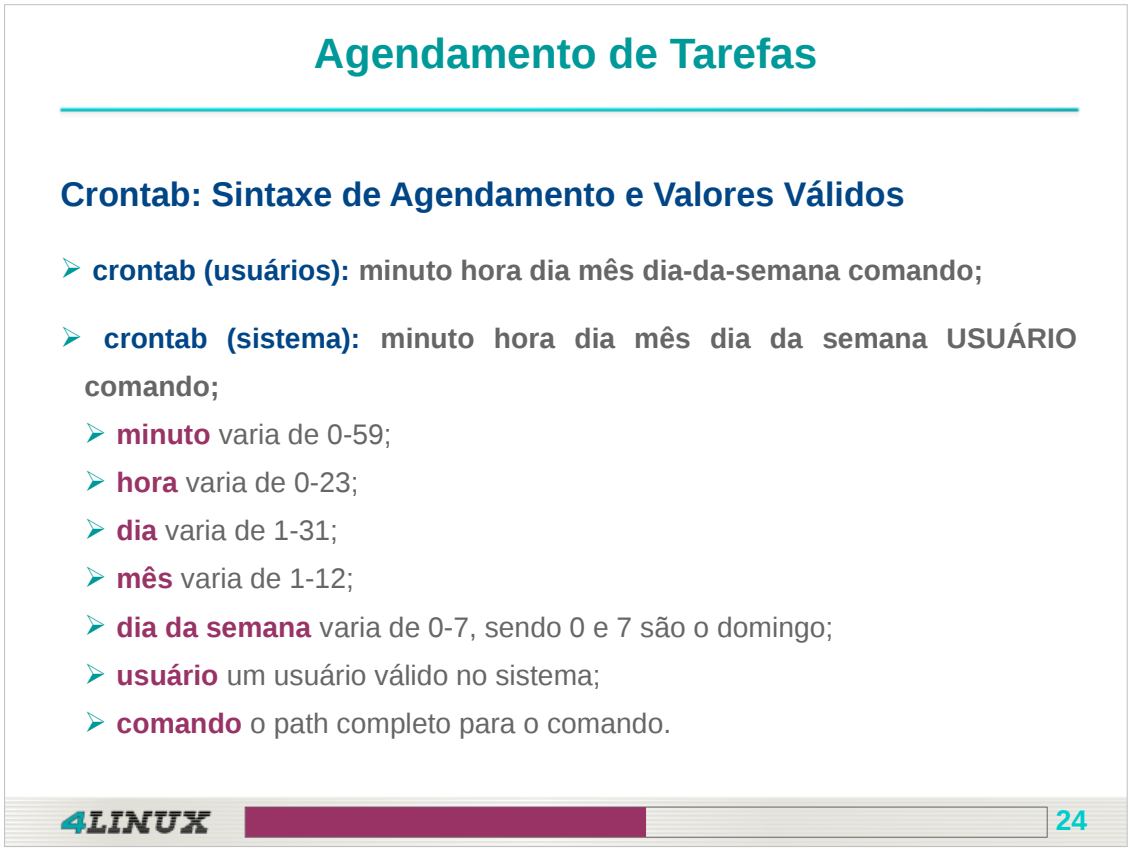
Servidor: WebServer Interno

- 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;
/etc/cron.d/ → Diretório que poderá conter outras tabelas de tarefas agendadas.

Anotações:



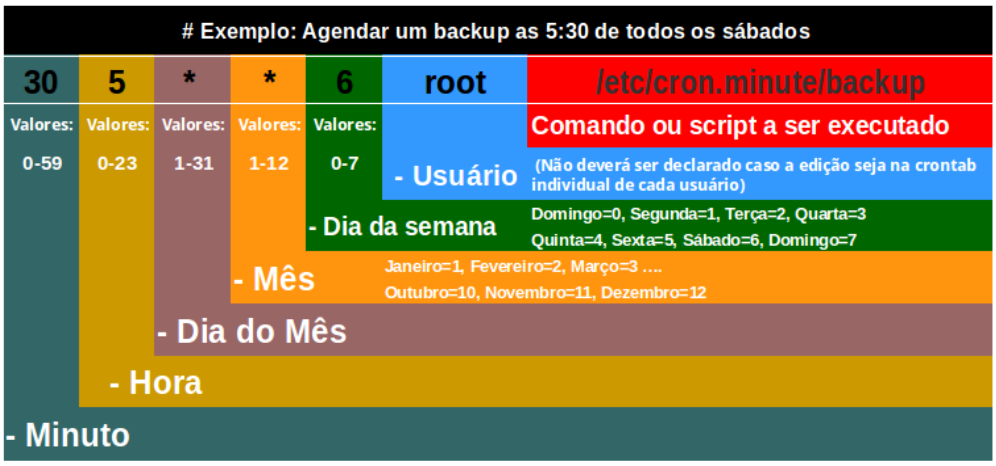
Cuidados ao elaborar scripts para execução pelo cron:

Scripts agendados não podem conter comandos cuja execução requer interação com o usuário, tal como pedir senha para completar a conexão de um ftp, ou comandos que requerem confirmação para execução como por exemplo apagar arquivos (O comando rm costuma pedir confirmação, a não ser que seja adicionado o parâmetro -rf).

Nestes casos todas as instruções devem ser colocadas dentro do script de maneira que possa completar a conexão passando o login e senha.

Agendamento de Tarefas

Resumindo:



Cuidados ao elaborar scripts para execução pelo cron:

Fique atento às permissões, quando possível agende como root para executar o script, use o "sudo"para dar permissões de execução em programas que requer poderes de root na execução, acesso a diretórios de backup e etc....

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".

Anotações:

Agendamento de Tarefas

Servidor: WebServer Interno

Crontab do Sistema:

Criaremos um pequeno script para validar o funcionamento da crontab do sistema, localizada em **/etc/crontab**:

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

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

```
#!/bin/bash
tar -cjf /tmp/bkp_home_$(date +%H\:%M).tar.bz2 /home
echo "Backup Realizado com sucesso!" | wall
```

2 – Conceda permissão de execução a tarefa:

```
2# chmod +x /tarefa.sh
```


Anotações:

Agendamento de Tarefas

Servidor: WebServer Interno

Crontab do Sistema:

1 – Verifique a data com o comando date, em seguida abra a crontab do sistema e crie um agendamento para execução em 3 minutos:

```
1# date
2# vim /etc/crontab
m h * * * root /tarefa.sh
```

< Marque a execução para daqui a 3 minutos >

Anotações:

Agendamento de Tarefas

Servidor: WebServer Interno

Crontab do Sistema:

2 – Para acompanhar o funcionamento do cron abra um segundo terminal e digite:

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

3 – Após 2 minutos verifique o diretório tmp:

```
2# ls /tmp/
```

Por padrão a execução de tarefas pelo cron não trás retorno em tela.

Por isso para acompanhar nossas execuções utilizamos o comando tailf que permite ver o final, “calda” de um arquivo em tempo real, neste caso o arquivo de log do cron.

Anotações:

Agendamento de Tarefas

Servidor: WebServer Interno

Crontab de Usuário:

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

1 – Verifique a crontab do usuário suporte:

```
1# crontab -lu suporte
```

no crontab for suporte

2 –Vamos criar a crontab para suporte:

```
2# su – suporte
3# crontab -e
```

Opções do crontab:

-e→ Edita o agendamento;

-r→ Remove o agendamento;

-l→ Lista agendamentos de um usuário;

-u→ Especifica o nome do usuário a ser gerenciado.

Anotações:

Agendamento de Tarefas

Servidor: WebServer Interno

Crontab de Usuário:

3 – Adicione a seguinte tarefa:

```
*/3 * * * * /tarefa.sh
```

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

```
1# exit
2# crontab -lu suporte
3# ls /tmp
```

5 – Apague a crontab do usuário

```
4# crontab -ru suporte
```

NOTA: Outra forma de visualizar os agendamentos é através dos arquivos encontrados no diretório /var/spool/cron/*

```
cat /var/spool/cron/suporte
```


Anotações:

Agendamento de Tarefas

Crontab: Outros Parâmetros Possíveis

Alguns parâmetros poderão ser utilizados para substituir os 5 primeiros campos referentes a data e hora de execução do agendamento:

@reboot → Especifica execução quando a máquina for reiniciada;
@yearly ou **@annually**→ Especificam execução uma vez ao ano, equivale a 0 0 1 1 *
@monthly → Especifica execução somente uma vez ao mês, equivale a 0 0 1 * *
@weekly → Especifica execução somente uma vez ao mês, equivale a 0 0 * * 0
@daily ou **@midnight**→ Especifica execução uma vez ao dia, equivale a 0 0 * * *
@hourly → Especifica execução somente uma vez a cada hora, equivale a 0 * * * *

Anotações:

Agendamento de Tarefas

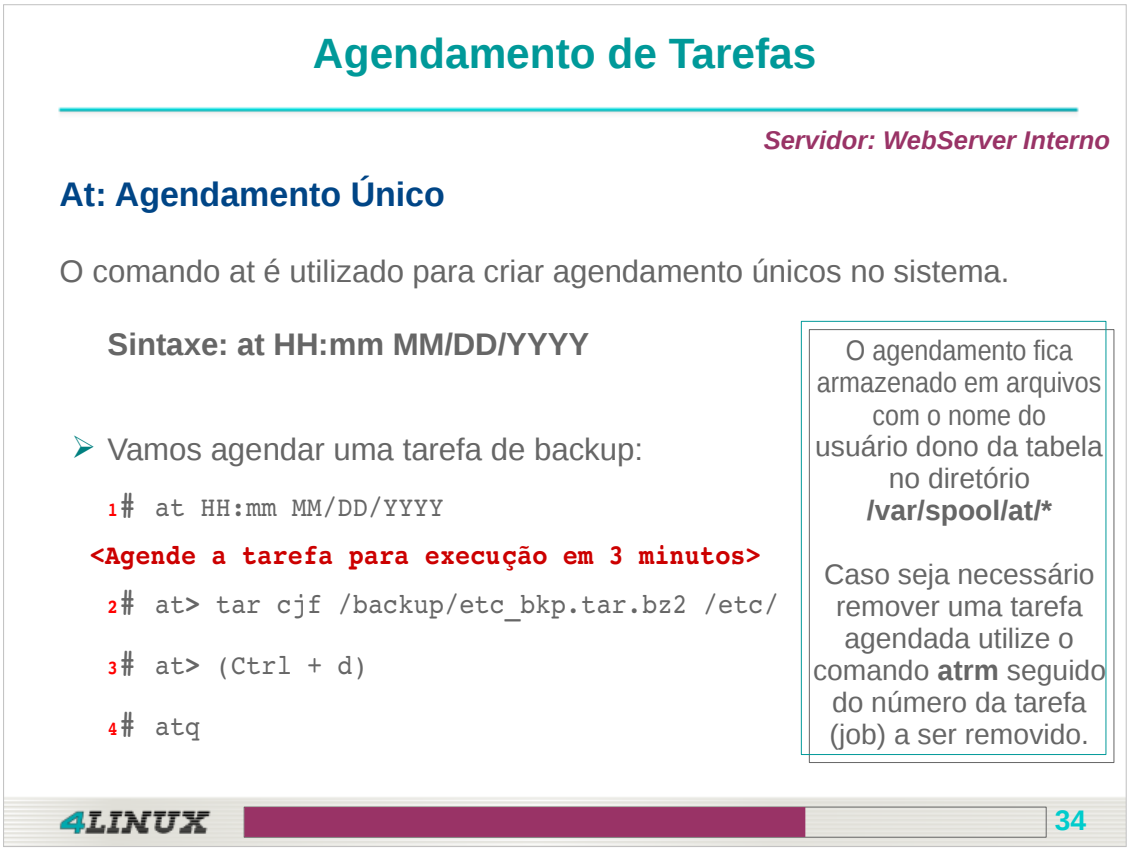
Anacron:

O conceito de funcionamento dos agendamentos de tarefas via cron pode ser aplicado a um outro agendador de tarefas chamado **anacron**; Este agendador possui a função de executar tarefas que por algum motivo não foram executadas, geralmente pelo fato da máquina estar desligada no momento agendado para execução.

1 – No arquivo /etc/anacrontab é possível verificar as especificações de agendamentos do anacron:

```
1# cat /etc/anacrontab
```


Anotações:



Agendamento de Tarefas com AT

O comando "at" pode agendar tarefas de forma semelhante ao cron, e é integrado à interface de linha de comando do Linux. É muito eficiente se aplicado no agendamento de tarefas que sejam disparadas somente uma vez. O at permite o controle dos usuários que podem agendar comandos através dos arquivos /etc/at.allow e /etc/at.deny.

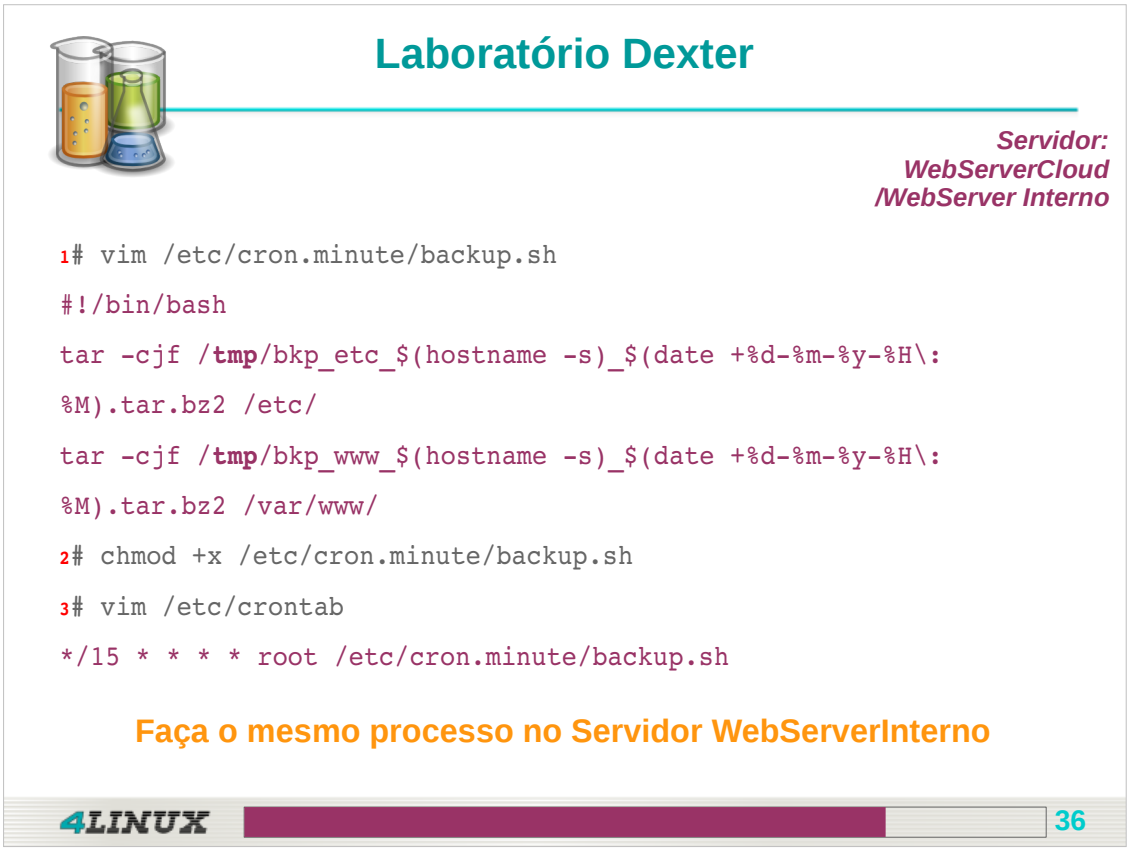
Estes arquivos são organizados no formato de um usuário por linha. Durante o agendamento é verificado primeiro o arquivo /etc/at.allow (listando quem pode executar o comando) e depois /etc/at.deny.

Caso eles não existam, o agendamento de comando é permitido a todos os usuários.

Cuidados Especiais com Scripts de Backup

Utilize nos comandos do script e no agendador, sempre o (path) caminho completo do aplicativo a ser executado, exemplo para o comando tar, use /bin/tar, também na linha de comando que inserir no cron use o caminho completo para o script, por exemplo, executar um script que está em /home/zago, use a linha: /home/zago/nome-do-script e não somente nome-do-script.

Muito cuidado com scripts, o comodismo pode cair no esquecimento e não atualizar o script de backup quando incluir novos serviços, diretórios ou usuários, monitore constantemente, teste e avalie o que está sendo feito.



Cuidados Especiais com Scripts de Backup

Fique atento às permissões, quando possível agende como root para executar o script;

Use o "sudo" para dar permissões de execução em programas que requer poderes de root na execução, acesso a diretórios de backup e etc....

Restringindo o Uso de Crontab

Servidor: WebServer Interno

- 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";
- Esse controles são lidos cada vez que o usuário tenta adicionar ou apagar uma tabela do cron.

Anotações:

Restringindo o Uso de Crontab

Servidor: WebServer Interno

- Restringindo o acesso do usuário suporte ao cron:

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

```
< Adicione uma linha como o nome suporte >
```

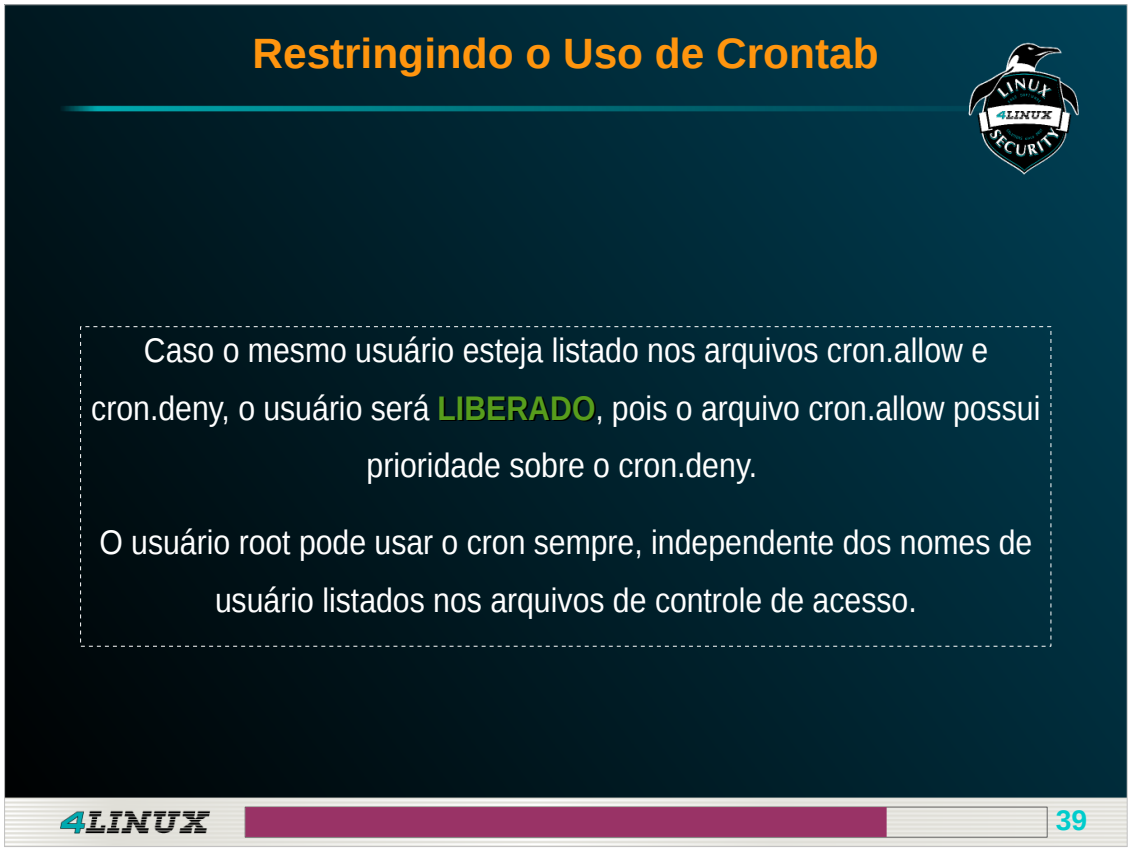
- Suporte:

```
< 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 >
```


Anotações:



Anotações:

Pergunta LPI

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

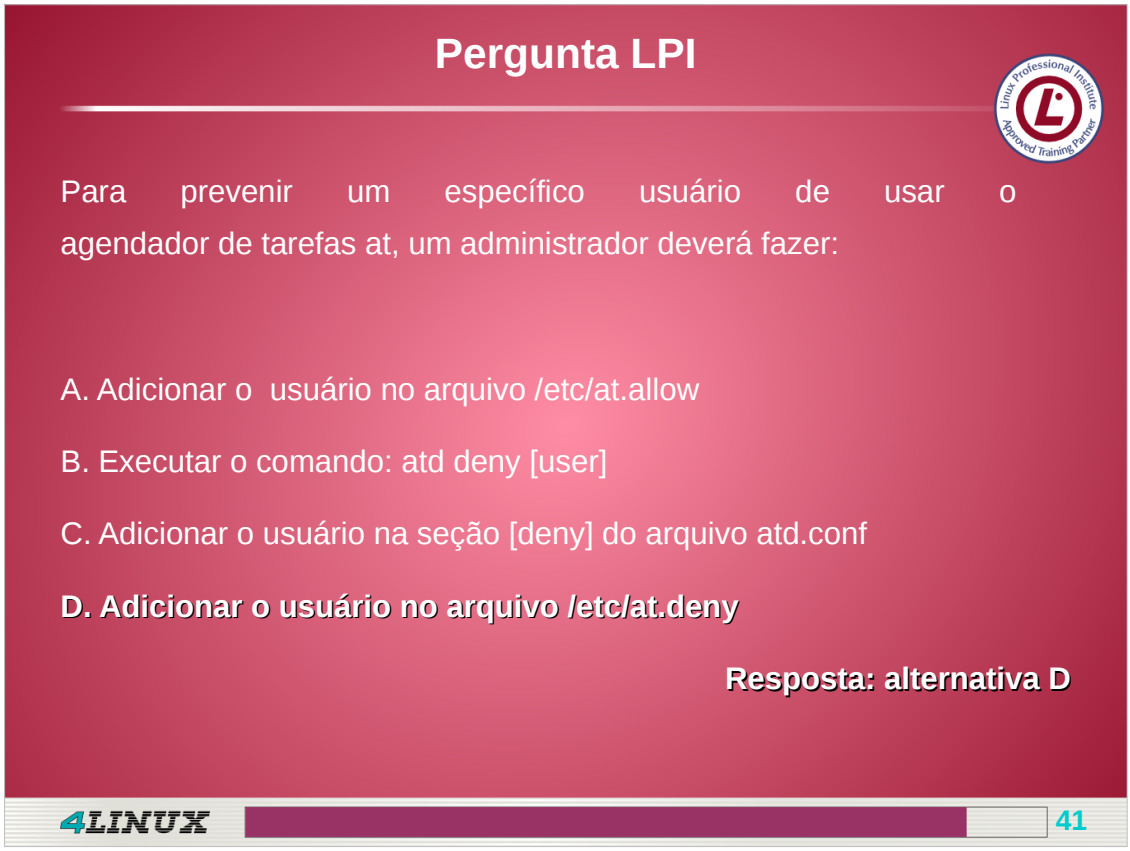
A. Adicionar o usuário no arquivo /etc/at.allow

B. Executar o comando: atd deny [user]

C. Adicionar o usuário na seção [deny] do arquivo atd.conf

D. Adicionar o usuário no arquivo /etc/at.deny

Anotações:



Alternativa: D RESPOSTA CORRETA!

Para bloquear o acesso de um usuário ao **at** basta adicionar o nome do usuário em questão no arquivo **/etc/at.deny** utilizando o mesmo formato adotado no bloqueio de acesso ao cron.

DICA:

O arquivo **at.allow** possui prioridade sobre o arquivo **at.deny,** perguntas simulando cenários onde ambos os arquivos recebem um mesmo nome de usuário podem ser cobradas na LPI!

Pergunta LPI

Quais das afirmações abaixo sobre o crontab são verdadeiras? (Escolha duas opções)

A. Todo usuário pode possuir a sua própria crontab

B. Ao modificar uma contrab uma reinicialização do daemon cron será requerida

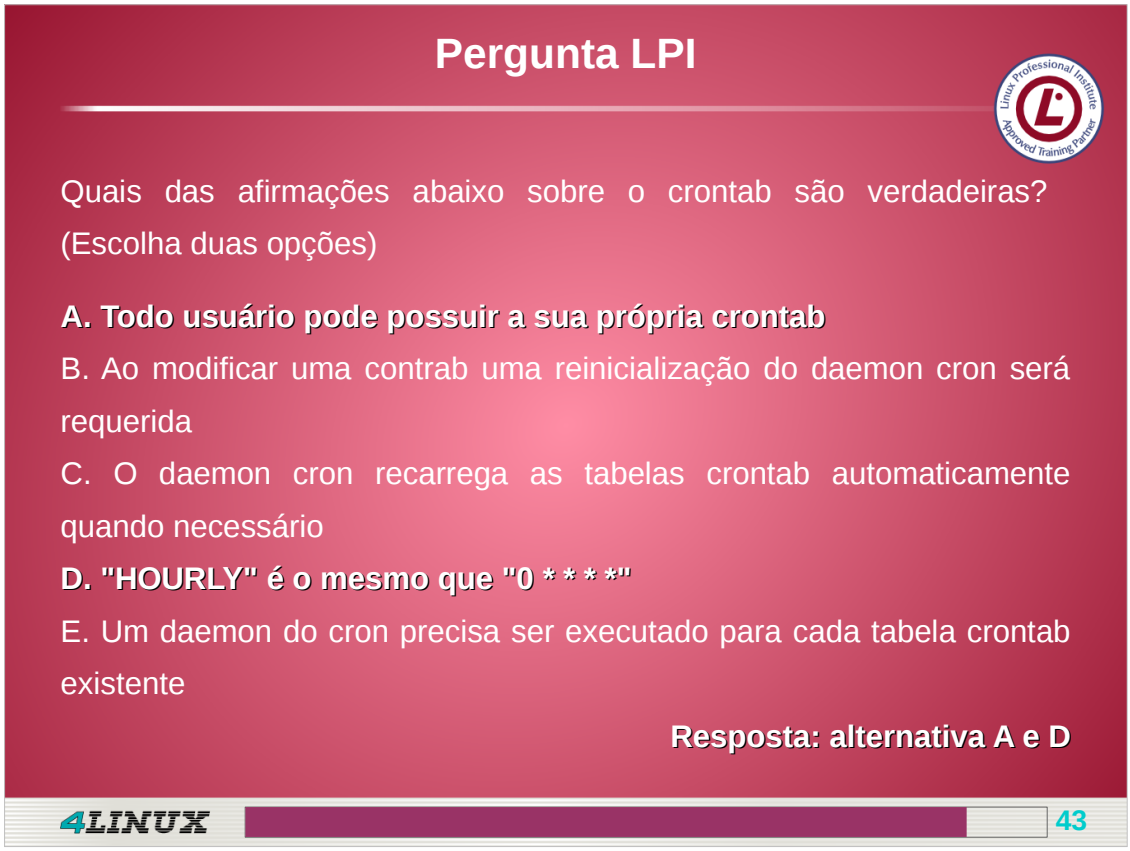
C. O daemon cron recarrega as tabelas crontab automaticamente quando necessário

D. "HOURLY" é o mesmo que "0 * * * *"

E. Um daemon do cron precisa ser executado para cada tabela crontab existente

Resposta: alternativa A e D

Anotações:



Alternativa A: RESPOSTA CORRETA!

Todo usuário pode possuir sua próṕria conrtab, essas tabelas podem ser editadas através do comando crontab e dos parâmetros descritos em aula.

Alternativa D: RESPOSTA CORRETA!

A string **“hourly”** representa o mesmo período de execução que o padrão **“0 * * * * ”** quando escrito por extenso.

Próximos Passos

Para que você tenha um melhor aproveitamento do curso, participes das seguintes atividades disponíveis no Netclass:

- Executar as tarefas do **Practice Lab**;
- Resolver o **Desafio Appliance Lab** e postar o resultado no Fórum Temático;
- Responder as questões do **Teste de Conhecimento** sobre o conteúdo visto em aula.

Mãos à obra!

Anotações:

Anotações: