

O slackware é a distribuição linux mais antiga ainda em atividade. Tendo sido criada por Patrick Volkerding em 1993, a partir da SLS.

Em todos esses anos, a distro conquistou ardorosos utilizadores, principalmente graças à sua filosofia de simplicidade e estabilidade.

Um produto de extrema qualidade para usuários com esta mesma característica. E este zine é de slacker para slacker.



slackware zine

Slackware is a registered trademark of Slackware Linux, Inc.

20 de Junho de 2015 - Edição #18

Editorial

É... Já se passaram 11 anos, mas parece que foi ontem...

Vamos ficando mais velhos. Com mais responsabilidades. E vamos deixando projetos e sonhos um pouco empoeirados na estante. Porém, não por ter menos apreço ao leitor da Slackware Zine.

Não podemos mais por todas essas "coisas de adultos" manter uma edição bimestral como no passado agora um pouco distante, mas manteremos uma edição anual, provavelmente coincidindo com o mês do Slackware Show.

Queremos agradecer mais uma vez aos nossos colaboradores, que reservam um tempo e dedicação pra nos presentear com um pouco de seu conhecimento e pesquisas.

Nesta edição trataremos basicamente de automação de ambientes de TI. Gerenciamento de pacotes, gerenciamento de configuração.

E também é com prazer que queremos anunciar que a partir desta edição da Slackware Zine estaremos iniciando as SlackDicas. Que nada mais são que superdicas para você leitor continuar um sysadmin assim... mais slacky.

Lembramos que estamos sempre prontos para receber críticas e opiniões para melhoria da qualidade da zine.

Have Fun!!

Eliane Vidal (mace)

índice

Docker poderia substituir o gerenciador de pacotes?

Rafael Gomes

2

PUPPET: Solução de gerenciamento de configuração

Eliane Vidal

3

SlackDicas: Usando Shell Like a Pro

Rodrigo Mosconi

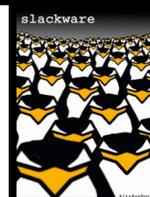
4

Reprodução do material contido nesta revista é permitida desde que se incluam os créditos aos autores e a frase:

"Reproduzida da Slackware Zine #18 -

www.slackzine.com.br"

com fonte igual ou maior à do corpo do texto e em local visível



slack
users

Docker poderia substituir o gerenciador de pacotes?

A utilização de containers, que tem hoje o Docker como principal representante, tem aumentado rapidamente com sua facilidade e portabilidade de liberação de aplicações prontas. Com base nisso, fica a seguinte pergunta: “Poderia a tecnologia de containers, como o Docker, ser utilizada para resolver o nosso antigo problema com gerenciador de pacotes? Alguns acreditam que containers podem ser um caminho para resolver a “dor de cabeça” no gerenciamento de dependência e incompatibilidade entre distribuições e versões do mesmo sistema.



Gerenciamento de pacotes no GNU/Linux sempre foi uma dor de cabeça para algumas situações.

No campo do “sim”: O pessoal do CoreOS, que são criadores de uma distribuições GNU/Linux, que gira inteiramente em torno de containers, e não dos pacotes, como unidade básica de modularidade.

Kelsey Hightower, Engenheiro Chef do CoreOS, diz que a empresa oferece “prova viva que não é apenas possível, como pode lidar melhor com designer de sistema e eficiência.” Contudo ele acredita que gerenciador de pacotes ainda é utilizável, “principalmente para construção de um sistema operacional a partir de um conjunto discreto de componentes que devem executar em conjunto. Esta é uma área onde os gerenciadores de pacotes do GNU/Linux brilham.” CoreOS não utiliza um gerenciador de pacotes típico, como apt ou yum, mas sim um sistema portage.

Kelsey alega que substituir inteiramente um

sistema de gerenciador de pacotes de propósito genérico para o Docker ou outro gerenciador de container pode ser difícil, em parte porque o Docker não tem mecanismo para solucionar dependência. “Onde Docker brilha é no empacotamento e distribuição de aplicações”, ele diz. (Docker se recusou a responder esse artigo, alegando restrição de tempo.)

Com ênfase maior da Red Hat em containers, é lógico que a empresa também poderá ver containers como um substituto para o gerenciamento de pacotes.

Lars Herrmann, gerente geral do “Red Hat Enterprise Linux” e “Red Hat Enterprise Virtualization” na Red Hat, acredita que é possível que o gerenciador de pacote seja substituído por containers, “mas esse não é o melhor caminho”.

Além da gerência de dependência, diz Herrmann em um email, que um gerenciador de pacote provem mais três outros “boons”: Instruções de onde será instalado o software em seu sistema, metadados estruturados para tornar fácil dizer o que e onde será instalado, e um mecanismo para verificar softwares instalados com base no metadado do gerenciador de pacotes.

Herrmann diz: “Estes valores aplicam-se ainda em um mundo ‘container-centric’ ” O que significa que todas essas funções precisam ser replicados pelos containers. “Além disso, é preciso haver uma maneira fácil de ‘olhar’ para dentro desses containers, afim de verificar o que está lá dentro, e assim identificar quaisquer problemas conhecidos.”

Da mesma forma, o gerenciamento de pacotes abrange funções internas que os próprios containers ainda não lidam. De acordo com Herrmann, o Docker “agrega os pacotes para uma aplicação ou microservice inteiro; isso não ajuda obter os componentes certos para um container de forma funcional. Obviamente, essa tarefa não requer diretamente os gerenciadores de pacotes existentes, mas eles fazem um grande trabalho, então por que não usá-los?

Bryant Cantrill, CTO da Joyent (outra empresa até

continua...

continua...

o pescoço no mundo do container) Também vê os containers e gerenciadores de pacotes servindo para funções separadas, desde que “uma imagem Docker está em uma camada de abstração mais elevado do que um pacote apt ou yum” Ele escreveu isso por e-mail.

Dito isso, ele acredita que o Docker poder tornar o empacotamento obsoleto, como o gerenciamento de pacote fez obsoleto a forma manual de desempacotar arquivos (como arquivos TAR) e assim se tornar um novo padrão de formato para aplicações. “Um engenheiro que conheço a muito tempo, coloca o Docker de forma diferente: Docker é ELF do século 21. (ELF é o ‘Executable and Linkable Format’ que é como o binário é representado em sistemas UNIX) De qualquer maneira, Docker parece prestes a se tornar o padrão de fato para a forma como se cria imagem para sistemas. Eu acho que isso é uma clara vitória tanto para desenvolvedores, como para os operadores de infraestrutura.”

Esse artigo é uma tradução livre do artigo “Could Docker replace package management?” da Infoworld. Encontrei esse artigo e achei interessante para levantar esse debate. Comentem, o que acham disso? Realmente o Docker, ou qualquer outro gerenciador de containers pode ser um sucessor do gerenciador de pacotes?

Containers é um bom caminho para “tornar” obsoleto o gerenciamento de pacotes, mas ainda não está pronto. As duas soluções vão coexistir durante algum tempo, mas realmente acho que o nível de abstração subirá um pouco mais em médio prazo, pois não acredito que teremos tempo para resolver manualmente instalações de dependência para soluções complexas.

Assim como hoje é possível compilar os pacotes manualmente, acho que no futuro continuaremos podendo instalar “da moda antiga”, mas o uso da solução de containers prontos, customizáveis e portáteis será quase uma necessidade. Talvez isso faça o conceito de distribuição perder relevância, mas é algo que ainda não tenho como certo em minha reflexão ainda e fica para outro texto.

Rafael Gomes
<claytones@terra.com.br>

PUPPET: Solução de gerenciamento de configuração

Solução homologada para os seguintes ambientes: Linux/Unix, Windows e Mac OSX.

Objetivo: Centralizar e organizar, criando uma estrutura padronizada e centralizada de configuração. Facilitando o gerenciamento de ativos. Agiliza também a criação de inventários de software.

O que é o Puppet?

Sistema Open source de automação de infraestrutura de hosts e serviços.

O Puppet faz uso de linguagem declarativa para expressar as configurações que serão aplicadas nos clientes.

Como trabalha?

O servidor (master) mantém uma base (catálogo) com as configurações que são aplicadas nos clientes.

Os clientes, por sua vez, rodam o agente que lê o catálogo e executa as configurações a ele destinadas e envia relatórios para o servidor.

Vantagens do uso do Puppet:

- => Centraliza o gerenciamento dos hosts e serviços
- => Reduz custos com manutenção de ativos
- => Diminui tempo de indisponibilidade derivadas de mudanças no ambiente
- => Menor tempo para manutenção de diversos hosts, deixando as equipes mais livres para resolverem outros problemas
- => Possibilitam testes de configuração
- => Deploy de aplicações em larga escala

Interfaces de gerenciamento do banco:

A interface de gerenciamento padrão do PuppetDB (base de dados do Puppet) é o Puppet Dashboard. Porém, é usual verificar o uso o PuppetBoard como alternativa de gerenciamento Web para o mesmo.

Enfim, para os que usam o Salt, vale mencionar que, a pesar de o Salt ser uma solução mais simples para este tipo de solução, o Puppet é uma solução mais robusta e maleável.

continua...

Links relevantes:

<https://puppetlabs.com/>
puppet-br.org/

IRC: [#puppet-br](irc://irc.freenode.net)

Eliane Vidal
<evidal@linuxajuda.com>

SlackDicas: Usando Shell like a Pro

Interfaces gráficas e/ou aplicações web que visam facilitar ou simplificar o trabalho de um sysadmin, já não são mais novidades para ninguém. Porém quando eventos fogem da zona de conforto dessas interfaces, o administrador é forçado a resolver problemas pela interface de linha de comando. Há quem diga que essa forma de trabalhar é arcaica, ou mesmo lenta, mas um administrador competente e experiente mostra-se

mais produtivo na boa e velha linha de comando do que em interfaces ditas modernas. Storages, roteadores ou switches modernos de qualidade possuem algum tipo de interface de linha de comando. Esta interface é fundamental para automatização de tarefas.

Na família Unix, e seus derivados ou inspirados, são denominadas da Shell as interfaces que fazem interação com o usuário para a execução de programas. Nesta definição, o Gnome, KDE, entre outros Window Managers são também Shell, porém o termo Shell normalmente está relacionado às interfaces de linha de comando como o Bash, Tcsh Pdksh ou o Zsh. Este artigo visa mostrar aos leitores recursos do Shell que podem auxiliar o seu dia a dia. O shell de linha de comando deverá se tornar o seu principal ambiente de trabalho. Logo este não aborda o Shell script, mas sim como usar o Shell como um ambiente de trabalho. Entre os Shells supra mencionados, escolha um, por exemplo o Bash (visto este é o padrão em vários ambientes Linux).

Recomendo a execução dos seguinte comandos:

```
$man -t bash | ps2pdf - bash.pdf
```

Pronto. Agora você tem um novo livro de cabeceira.

Movimentação no shell

Aprender a movimentar na linha de comando, sem precisar depender das teclas home/end ou das setas direcionais é altamente produtivo, principalmente quando estamos usando um teclado com algum layout diferente do habitual, ou de laptop, notebook ou netbooks, onde o espaço para uma disposição convencional é um artigo de luxo.

Além da parte herdada de uma máquina datilográfica, são fundamentais, e presentes no nosso universo de uso, a tecla CTRL e a tecla ESC ou a tecla Alt. As duas últimas implementam o mesmo comportamento, e durante este artigo será utilizada a tecla ESC. Caso algum comportamento varie entre as tecla ESC e ALT será devidamente notificado.

Sem HOME ou END, ou setas direcionais:

Aprender a não depender do home/end pode propiciar as seguintes vantagens para o sysadmin: a) redução da perda de tempo com movimentação das mãos e; b) redução de DORT.

Para retornar ao início da linha, basta usar Ctrl+A e para ir ao final, Ctrl+E.

Para avançar ou retornar exatamente 1 caracter usa-se Ctrl+F ou Ctrl+B, respectivamente.

Podem-se memorizar com os seguintes mnemônicos:

Ctrl+A : A é o início do alfabeto, A é o início da linha

Ctrl+E : E de End of line

Ctrl+F : F de Forward

Ctrl+B : B de Backward

Mas mover de caracter em caracter pode não ser desejável ou produtivo. E mover para início, ou final, da linha também pode não ser interessante. Com Esc-F e Esc-B é possível avançar o retornar uma palavra. Palavras são delimitadas por: espaço; caracteres de pontuação; barra; hífen ou underscore.

Nos vemos na próxima edição com: "Delete para quê?"

Rodrigo Mosconi
<rmosconi@linuxajuda.com>