

Configuração de Dispositivos de Rede Utilizando Templates

Rubens Kühl Jr.

LARC/USP

rubens@email.com

Configuração de Dispositivos de Rede

- Tipicamente baseada em linha de comando
- Em geral os comandos podem ser agregados em lotes e transferidos como arquivos de múltiplas linhas por TFTP, FTP ou SCP
- Isso permite tratar as configurações com estratégias de geração e tratamento de arquivos

Templates

- ***Template***

- “modelo, molde, padrão, gabarito.”
(Michaelis)
- Arquivo que serve como ponto de partida para um novo documento
- Preenchem-se características específicas, padrão geral é dado pelo template

- Podem ser aplicados à configuração de dispositivos de rede

O Desafio

- Configurações de portas, interfaces, filtros, e rotas em switches, roteadores e load-balancers numa rede de missão crítica (em disponibilidade e segurança da informação)
- 1 dúzia de modelos de dispositivos, papéis diferentes na rede, quase 50 dispositivos no total
- Desses, 4 modelos de dispositivos, 7 papéis diferentes na rede (14 dispositivos) com grande número de linhas de configuração receberam esforços de automação

A Resposta

- Template Toolkit
 - <http://www.template-toolkit.org/>
- Ferramenta open-source, baseada em Perl na sua construção e em extensões sintáticas
- Apesar do grande uso em páginas Web, agnóstico a tipo de arquivo

O Resultado

- 4 modelos, 7 papéis, 14 dispositivos
- 414 arquivos de configuração
- 2027 linhas de configuração (muitas geradas por copy/paste de arquivos similares)
- 28954 linhas geradas pelo processador de templates

Estratégia de construção

- Um template inicial para cada papel de equipamento
- Um template de informações individuais de cada equipamento
- Arquivos de cada ítem de configuração, que apontam para os templates anteriores
- Cada equipamento, um diretório

Alguns exemplos

- Arquivo de definição de uma access-list

```
[% USE file(template.name, nostat=1) %]
```

```
[% acl = file.name FILTER remove '\.tt2$'  
%]
```

```
no ip access-list extended [%acl%]
```

```
ip access-list extended [%acl%]
```

```
... Comandos ....
```

```
exit
```


Alguns exemplos

- Arquivo de informações de um dispositivo

```
[% interligips = [ '172.27.255.50'  
  '172.27.252.50' ] %]
```

```
[% routerips = [ '172.27.1.3' '172.27.1.1'  
  '172.27.2.3' '172.27.2.1' '172.27.3.3'  
  '172.27.3.1' '172.27.4.3' '172.27.4.1'  
  '172.27.5.3' '172.27.5.1' '172.27.9.3'  
  '172.27.9.1' '172.27.13.3' '172.27.13.1'  
  '172.27.14.3' '172.27.14.1' '172.27.19.3'  
  '172.27.19.1' ]; routerips =  
routerips.merge(interligips) %]
```

Alguns exemplos

- Arquivo de configuração

```
[% ips = [ '172.27.3.20', '172.27.3.50'  
  ] %]
```

```
[% interface = xx %]
```

```
[% PROCESS 'papelX-1-info.tt2' %]
```

```
[% PROCESS 'papelX-acl-in.tt2' %]
```

Ganhos

- Menor tempo de implantação
- Drástica redução de erros (e que nem sempre seriam descobertos)
- Agilidade de adaptação; templates de construção mudaram bastante em mais de um momento, em função de problemas e limitações
- Inteligibilidade e auditabilidade da configuração

Possibilidades

- Utilização da integração do TT (Template Toolkit) com banco de dados (de .dbm a SQL)
- Geração automatizada dos templates (template do template)
- Criação de makefile para caracterização de dependências