

4525 – Curso de Infraestrutura Ágil com Práticas DevOps

Desafio - Laboratório Parte 3: Automação PostgreSQL e Memcached.

Vamos utilizar o conhecimento que tivemos no desenvolvimento no laboratório da Dexter **Parte 1 – Automação com DNS, Parte2 – Automação do Apache e Nginx e Parte 3 – Automação PostgreSQL e Memcached** para melhorar o nosso ambiente de teste automatizados. Na Aula de Automação com Ansible nos desenvolvemos um Playbook para realizar o Deploy do módulo do puppet dentro do servidor puppet master e na aula Gerência de Configuração com Puppet nos vimos o comando **puppet parser validate**.

Parte 1. Melhorando teste automatizados do ambiente da Dexter.

Nesse laboratório vamos alterar essa Playbook que na verdade é uma role denominada puppet (/etc/ansible/roles/puppet) para realizar o comando puppet parser validate nos arquivos de manifests do módulo do puppet, então nesse novo ambiente de teste precisamos executar um puppet parser validate nos arquivos **manifests/files.pp**, **manifests/install.pp**, **manifests/service.pp** e **manifests/init.pp** e para não prejudicar o processo de automação só podemos executar o comando nos arquivos caso o arquivo exista. sendo assim o módulo de DNS por exemplo ira passar nos teste sem problemas de configuração, pois não possui os arquivos de manifests de services, install e files porém sua configuração está escrita de forma correta.

Esse teste deve ser feito logo após o deploy dos arquivos no servidor, ao realizar esta configurações evitamos muitos problemas que poderíamos enfrentar caso o agente do puppet for configurado com falha nas sintaxes, por exemplo o container ser criado porém o puppet não aplicou as configurações, e container está com redirecionamentos de porta para a máquina Docker Host.

Dica: existe um comando no linux que podemos utilizar para ver se o arquivo existe: `test -f <path/to/file>`, caso esse comand for executado com sucesso, você precisa realizar o `puppet parser validate`.

No arquivo: `/etc/ansible/roles/puppet/tasks/main.yml` foi adicionado o `include` para o arquivo `test.yml` sendo assim deixando o arquivo mais limpo e fácil para gerenciar (o código adicionado está em **destaque** abaixo):

- name: Update puppet modules

git:

repo: "{{ repo }}"

dest: /etc/puppet/environments/{{ env }}/modules/{{ modulename }}

key_file: /etc/keys/gitlab

- name: clean-up certificates

command: puppet cert clean {{ certificatename }}.dexter.com.br

when: certificatename is defined

ignore_errors: yes

- include: test.yml

No arquivo: `/etc/ansible/roles/puppet/tasks/test.yml` foi criado os teste sendo que antes de executar o comando `puppet parser validate` e validado se o arquivo existe através do comando `test` e parâmetro `-f` para verificar se o arquivo existe ou não, caso o arquivo exista o retorno do comando é valor 0 caso o arquivo não exista o retorno é valor 1.

===== TESTING INIT FILE =====

- name: testing if manifests init.pp file exists

command: test -f /etc/puppet/environments/{{ env }}/modules/{{ modulename }}/manifests/init.pp

register: result_init

ignore_errors: yes

- name:

command: puppet parser validate /etc/puppet/environments/{{ env }}/modules/{{ modulename }}/manifests/init.pp

when: result_init.rc == 0

===== TESTING INSTALL FILE =====

- name: testing if manifests install.pp file exists

command: test -f /etc/puppet/environments/{{ env }}/modules/{{ modulename }}/manifests/install.pp

register: result_install

ignore_errors: yes

- name:

command: puppet parser validate /etc/puppet/environments/{{ env }}/modules/{{ modulename }}/manifests/install.pp

when: result_install.rc == 0

===== TESTING FILES FILE =====

- name: testing if manifests files.pp file exists

command: test -f /etc/puppet/environments/{{ env }}/modules/{{ modulename }}/manifests/files.pp

register: result_files

ignore_errors: yes

- name:

command: puppet parser validate /etc/puppet/environments/{{ env }}/modules/{{ modulename }}/manifests/files.pp

when: result_files.rc == 0

```
# ===== TESTING SERVICE FILE ===== #  
- name: testing if manifests service.pp file exists  
  command: test -f /etc/puppet/environments/{{ env }}/modules/  
{{ modulename }}/manifests/service.pp  
  register: result_service  
  ignore_errors: yes  
  
- name:  
  command: puppet parser validate /etc/puppet/environments/{{ env }}/modules/  
{{ modulename }}/manifests/service.pp  
  when: result_service.rc == 0  
  
# END OF FILE
```