

## 4525 – Curso de Infraestrutura Ágil com Práticas DevOps

### Desafio - Laboratório Parte 2: Automação Apache e Nginx como Load Balancer

Vamos utilizar os conhecimentos utilizados no desenvolvimento da Segunda parte do Nosso laboratório Dexter, com a Automação do Serviço **Apache** e do Serviço **Nginx como Load Balancer** dos servidores do Apache para melhorar o nosso ambiente de teste automatizados configurados na Aula.

Quando o pacote do apache é instalado no sistema, ele traz um comando para gerenciar o apache chamado **apache2ctl** para distribuições Debian. O seu desafio e utilizar esse comando para realizar um **teste de configuração** aplicada pelo módulo do puppet. Sendo que durante a Aula nos fizemos um teste simples para verificar se serviço está rodando após a aplicação do agente do puppet.

Podemos também verificar se os arquivos que módulo garante estão localizados na máquina para garantir que o serviço seja configurado com sucesso, por exemplo podemos utilizar **`test -f <path/to/file>`** para verificar se o puppet criou o arquivo validando assim a configuração do serviço apache, através do módulo do puppet está a configuração do arquivos **`/etc/apache/apache2.conf`** e do diretório **`/etc/apache2/sites-available/`** e os arquivos que existem lá dentro. O seu desafio e criar teste para garantir que os arquivos foram criados com sucesso através do comando **`test`** que está presente nas distribuições linux por padrão, para garantir teste melhores é valido testar os arquivos de configuração, os diretórios de configuração e arquivos que estão dentro do diretório, **principalmente os arquivos do virtualhost do apache.**

**DICA:** Dentro do diretório do apache, realizar a criação de um arquivo chamado test.yml e realizar um include dentro do playbook main.yml para o serviço do apache no ambiente de homologação, sendo assim você terá uma organização das tarefas e facilitará a manutenção dos arquivos.

Para melhorar o gerenciamento dos arquivos, os arquivos foram modificados e criados mais dois arquivos:

/etc/ansible/playbooks/homolog/apache

- main.yml (Arquivo principal do playbook)
- docker.yml (Arquivo para provisionamento do container e chamada ao agent do puppet)
- docker-finish.yml (Arquivo para remover o container após realização do teste)
- test.yml (Arquivo com os testes a ser realizado para garantir que a configuração foi realizada com sucesso)

No arquivo: /etc/ansible/playbooks/homolog/apache/main.yml foi adicionado dois include's para os novos arquivos criados:

---

**# Playbook: Responsavel pelo servico do Apache**

**- name: provisionar o apache**  
**hosts: docker**

**tasks:**

- **include: docker.yml**
- **include: test.yml**
- **include: docker-finish.yml**

No arquivo: /etc/ansible/playbooks/homolog/apache/docker.yml foi os comandos para para

e remover o container:

---

**# Create container**

**- name: testing docker container**

**command: docker inspect node-homolog**

**register: container**

**ignore\_errors: yes**

**- name: stop docker container**

**command: docker stop node-homolog**

**when: container.rc == 0**

**- name: remove docker container**

**command: docker rm node-homolog**

**when: container.rc == 0**

**- name: create docker container**

**command: docker run -dit --name node-homolog --hostname node-homolog --add-host=puppet:192.168.200.50 homolog-template /bin/bash**

**- name: running puppet agent**

**command: docker exec node-homolog puppet agent -t --environment=homolog**

**ignore\_errors: yes**

No arquivo: **/etc/ansible/playbooks/homolog/apache/test.yml** foi os comandos para testar se os arquivos foram criados com sucesso:

---

**- name: testing apache configuration file**

**command: docker exec node-homolog apache2ctl configtest**

- name: testing configuration file exists

command: docker exec node-homolog test -f /etc/apache2/apache2.conf

- name: testing virtualhost directory exists

command: docker exec node-homolog test -d /etc/apache2/sites-enabled

- name: testing whether dexter virtualhost file exists

command: docker exec node-homolog test -f /etc/apache2/sites-enabled/dexter.conf

- name: testing apache configuration file

command: docker exec node-homolog service apache2 status

No arquivo: `/etc/ansible/playbooks/homolog/apache/docker-finish.yml` foi adicionado os comandos para parar e remover o container:

---

- name: docker stoping container

command: docker stop node-homolog

- name: docker removing container

command: docker rm node-homolog