



Agenda dia 03

Meu primeiro Manifesto

Meta-parâmetros e referência a recursos

Instrutor: Diogo A. M. Barbosa

↑ https://github.com/diogoab

diogo.alves.barbosa@gmail.com

whoami





Arquiteto de soluções na Everis Brasil, formado em Sistemas de Informação, a mais de 12 anos de experiência no mercado de TI atuando em áreas de infra e desenvolvimento de software, como DevOps utilizando ferramentas como Docker, Kubernetes, Ansible, Puppet, Chef, Jenkins, Cloud Computing com AWS, Azure, Google Cloud Platform, Infra com servidores Linux e Windows, desenvolvimento em PHP, Python e Ruby, possuo certificações como AWS Solutions Architect, Ethical Hacker dentre outras, apaixonado Linux e Open Sources e Instrutor de vários treinamentos como certificação LPI, Docker e DevOps.

Instrutor: Diogo A. M. Barbosa

https://github.com/diogoab

diogo.alves.barbosa@gmail.com







As declarações de configuração são chamadas de manifests (manifestos) e são armazenadas em arquivos com a extensão .pp

A principal utilidade da linguagem do Puppet é a declaração de resources, representando um estado desejado.

Nos manifests também podemos ter condições, agrupar resources, gerar texto com funções, utilizar e referenciar código em outros manifests e muitas outras coisas. Mas o principal é garantir que resources sejam declarados e gerenciados de maneira correta.

Exemplo 01

M diogo.alves.barbosa@gmail.com

```
# mkdir /root/manifests
   # vim /root/manifests/arquivo-1.pp
   file { 'teste':
     path => '/tmp/teste.txt',
     ensure => present,
     mode => '0640',
     content => "Conteudo de teste!\n",
 Instrutor: Diogo A. M. Barbosa
https://github.com/diogoab
```



Meu primeiro Manifesto # Exemplo 01

```
# puppet apply arquivo-1.pp
```

```
# cat /tmp/teste.txt
```

Conteudo de teste!

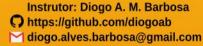


Meu primeiro Manifesto # Exemplo 02

```
# EXEMPIO UZ

# sudo vim /root/manifests/arquivo-2.pp
```

```
file { '/tmp/teste1.txt':
   ensure => present,
   content => "Ola!\n",
}
```

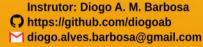




Exemplo 02 - Continuação

```
Meu primeiro Manifesto
```

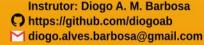
```
file { '/tmp/teste2':
 ensure => directory,
 mode => '0644',
```





```
# Exemplo 02 - Continuação
```

```
file { '/tmp/teste3.txt':
  ensure => link,
  target => '/tmp/teste1.txt',
notify {"Gerando uma notificação!":}
notify {"Outra notificação":}
```





Exemplo 03

```
# vim /root/rollback.pp
file{ 'remove_dir':
   path => '/home/maria',
   ensure => absent,
   force => true,
}
```



Exemplo 03 - Continuação

```
user{ 'remove user':
    name => maria,
    ensure => absent,
 group{ 'remove group':
    name => maria,
    ensure => absent,
 Instrutor: Diogo A. M. Barbosa
https://github.com/diogoab
M diogo.alves.barbosa@gmail.com
```









Agora que entendemos os manifests e a declaração de recursos, vamos aprender sobre meta-parâmetros e ordenação de recursos.

O neste encadeamento será utilizado os meta-parâmetros before, require, notify e subscribe.



Exemplo

```
file { '/tmp/teste1.txt':
  ensure => present,
  content => "Ola!\n",
file { '/tmp/teste2':
  ensure => directory,
 mode => '0644',
file { '/tmp/teste3.txt':
 ensure => link,
  target => '/tmp/teste1.txt',
notify {"Gerando uma notificação!":}
notify {"Outra notificação":}
```

Instrutor: Diogo A. M. Barbosa

https://github.com/diogoab
diogo.alves.barbosa@gmail.com



Prática 01

```
file { '/tmp/teste1.txt':
    ensure => present,
    content => "Ola!\n",
    }
notify { '/tmp/teste1.txt foi sincronizado.':
require => File['/tmp/teste1.txt'],
}
```



Prática 02

```
file { '/tmp/teste1.txt':
    ensure => present,
    content => "Ola!\n",
    before => Notify['mensagem'],
 notify {'mensagem':
    message => 'O arquivo testel.txt foi criado!',
 Instrutor: Diogo A. M. Barbosa
https://github.com/diogoab
M diogo.alves.barbosa@gmail.com
```



Prática 03

M diogo.alves.barbosa@gmail.com

```
file { '/tmp/teste1.txt':
   ensure => present,
    content => "Ola!\n",
 notify {'mensagem':
   require => File['/tmp/testel.txt'],
   message => 'O arquivo teste1.txt foi criado!',
 Instrutor: Diogo A. M. Barbosa
https://github.com/diogoab
```



Prática 03

```
file { '/etc/ssh/sshd_config':
    ensure => file,
    mode => '0600',
    source => '/root/manifests/sshd_config',
    notify => Service['sshd'],
}
```



Prática 03 - Continuação

```
service { 'sshd':
   ensure => running,
   enable => true,
   hasrestart => true,
   hasstatus => true,
}
```



Prática 04

```
file {'/tmp/teste1.txt':
  ensure => present,
  content => "Hi.",
notify {'depois':
  message => '/tmp/testel.txt foi sincronizado.',
File['/tmp/teste1.txt'] -> Notify['depois']
Instrutor: Diogo A. M. Barbosa
```



Exercício de fixação

Desenvolva uma manifesto que execute o seguinte:

Criar um usuário chamado mario

e criar os arquivos mario1.txt, mario2.txt no diretorio

/home/mario

Instale o apache2.



puppet

Até a próxima aula!