





Agenda dia 02

Preparar Ambiente

Manipulando Resources

Meu primeiro Manifesto

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





Preparar Ambiente Puppet



Requisitos - Lab 01

- → Conta Google Cloud;
- → Instância n1-standard-1 (1 vCPUj, 3,75 GB de memória);
- → Ubuntu-1604-xenial

Preparar Ambiente Puppet



Requisitos - Lab 01

```
$ sudo su
# mkdir /puppet
# wget http://apt.puppetlabs.com/puppetlabs-release-pc1-xenial.deb -P /puppet
# cd /puppet
# sudo dpkg -i puppetlabs-release-pc1-xenial.deb
# sudo locale-gen pt_BR.UTF-8
```

Instrutor: Diogo A. M. Barbosa

↑ https://github.com/diogoab

diogo.alves.barbosa@gmail.com

Preparar Ambiente Puppet



Requisitos - Lab 01

```
# export PATH=/opt/puppetlabs/bin:$PATH
# echo "PATH=/opt/puppetlabs/bin:$PATH" >> /etc/bash.bashrc
# echo "export PATH" >> /etc/bash.bashrc
# apt-get update
# sudo apt-get -y install puppet-agent
# puppet --version
```



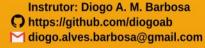


#HANDS'ON#

Manipulando Resources Puppet # Exemplo 01 - user



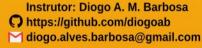
```
# sudo su
# puppet resource user
# puppet resource user root
```



Manipulando Resources Puppet # Resposta



```
user { 'root':
                 => 'present',
  ensure
  comment
                 => 'root',
 gid
                 => '0',
                 => '/root',
 home
 password
'$6$QAT/iZv0$KJuUEekD39uS9IG3n7x6GVWAGUPzH.Rbr2a27aPWqcY9bwaxJJwkX3MwY2VBAMOACdjNlJcLKPoGVi8wwC0PD0',
 password max age => '99999',
 password min age => '0',
  shell => '/bin/bash',
        => '0',
 uid
```



Manipulando Resources Puppet # Salvar arquivo manifest



```
# puppet resource user root > root.pp
```

passwd root





Comparar resultado

egrep root /etc/shadow



Testar e corrigir

```
# puppet apply root.pp --noop(no operation)
```

puppet apply root.pp



Criar usuario

```
# puppet resource user maria ensure=present
home="/home/maria" managehome=true
# id maria
uid=500(maria) gid=500(maria) groups=500(maria)
```



Exemplo 02 - service

```
# puppet resource service
# puppet resource service ssh
# puppet resource service ssh > sshd.pp
```

Manipulando Resources Puppet # Exemplo 02 - service

```
começando com puppet
```

```
# service ssh stop
# puppet apply sshd.pp --noop
# puppet apply sshd.pp
# service ssh status
```



Exemplo 02 - service

```
# puppet resource service ufw ensure=stopped
```

- # service ufw status
- # puppet resource service ufw ensure=running



Exemplo 03 - package

```
# dpkg -s aide
# puppet resource package aide ensure=installed
# dpkg -s aide
```



Atributos

puppet describe -s user

puppet describe user

Instrutor: Diogo A. M. Barbosa

https://github.com/diogoab

diogo.alves.barbosa@gmail.com



Modificando resources interativamente

```
# puppet resource user maria --edit
```

puppet describe user

Instrutor: Diogo A. M. Barbosa

↑ https://github.com/diogoab

diogo.alves.barbosa@gmail.com



Exemplo 04 - user

```
# puppet resource user joe ensure=absent
# puppet resource file /home/joe ensure=absent
force=true
```



Modificando resources interativamente

Adicione nova linha groups ['bin', 'adm'], #<-- essa linha é nova!





#HANDS'ON#

Meu primeiro Manifesto



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.

Meu primeiro Manifesto

começando com puppet

Exemplo 01

```
# 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
```



puppet

Até a próxima aula!