

## 4525 – Curso de Infraestrutura Ágil com Práticas DevOps

### Desafio – Versionamento com GIT

Neste desafio iremos versionar a configuração do diretório do **ansible** e do **puppet**, além de criar o repositório 'locais' do GIT para os arquivos de configurações, vamos aprender a trabalhar com commits e tags. De acordo com Infraestrutura da Dexter, cada módulo do puppet será versionando e terá um repositório no gitlab, então temos que procurar uma forma de ignorar o versionamento dos modulos do puppet dentro do diretório de configuração do puppet para evitar informações duplicadas no repositório do Git.

### Vamos versionar com GIT os arquivos de configuração do Ansible

1. Após o termino da configuração da Aula – Automação com Ansible, crie um repositório do GIT dentro do diretório **/etc/ansible**.

```
# cd /etc/ansible
```

```
# git init
```

2. Realize o commit de todos os arquivos que estão dentro desse diretório (/etc/ansible).

```
# git add -all
```

```
# git status
```

```
# git commit -m "Configuração do Ansible Aula03"
```

3. Crie uma Tag para o commit realizado anteriormente com o nome v1.0 e com uma descrição “v1.0 – Stable Release”.

```
# git log --oneline
```

```
# git tag -a v1.0 -m "v1.0 – Stable Release"
```

```
# git tag
```

## Agora, Vamos versionar com GIT os arquivos de configuração do Puppet

1. Após o termino da configuração da Aula – Gerência de Configuração com Puppet, crie um repositório dentro do diretório **/etc/puppet**.

```
# cd /etc/puppet
```

```
# git init
```

2. No ambiente da Dexter, precisamos encontrar uma forma de ignorar os arquivos que estão os módulos do puppet, que estão configurados no diretórios

**/etc/puppet/environments/homolog/modules** e

**/etc/puppet/environments/production/modules**. Então precisamos ignorar todos os arquivos que estão dentro desse diretórios garantindo que não tenha arquivos duplicados no nosso repositório interno da Dexter.

```
# vim .gitignore
```

```
environments/production/modules/
```

```
environments/homolog/modules/
```

```
# git status
```

Ao usar o arquivo `.gitignore`, você coloca quais arquivos e diretórios serão ignorados pelo GIT, sendo assim o git não gerencia as versões e não armazena em seu repositório os arquivos. Dentro do arquivo `.gitignore` você coloca o diretório que no nosso caso só queremos ignorar o diretórios do módulos de cada ambiente, lembrando que é apenas os módulos e não os arquivos de manifests como por exemplo: `environments/production/manifests/site.pp`. Outra informação importante é que os diretórios e arquivos são escritos tendo a pasta do repositório como raiz, que no caso `/etc/puppet`, como o repositório está dentro do `/etc/puppet`, dentro do `gitignore` basta colocar `environments/production/modules/`.

Link da documentação do gitignore: <https://git-scm.com/docs/gitignore>

3. Após a configuração do arquivo para ignorar os dois diretórios, realize o commit no repositório criado.

```
# git add -all
```

```
# git commit -m "Configuração do Puppet – Aula04"
```

4. Adicione uma tag ao commit anterior com o nome `v1.0` e descrição `"v1.0 – Stable Release"`.

```
# git tag
```

```
# git tag -a "v1.0" -m "v1.0 – Stable Release"
```

Após a conclusão da tarefa acima o seu desafio estará completo.