

01

Verificando o status HTTP

Transcrição

Os diretores da *Multillidae* nos disseram sobre o servidor da Apache que tem o conteúdo web, sendo ela a parte de acesso do sistema da empresa. Eles comentaram também que esse servidor é muito crítico. Por isso, eles nos pediram para encontrarmos uma forma do nosso script monitorar o **status** do Apache, a fim de verificar se ele está funcionando apropriadamente. Caso não esteja, o script deve **reiniciar o servidor**, e mandar um e-mail para o administrador do sistema, para que ele possa entender o que pode ter ocorrido com o servidor.

Para realizarmos essa tarefa, teremos que fazer o [download \(https://drive.google.com/open?id=0BzmYQVmw4W7nd1o2bFlhTmtLSWM\)](https://drive.google.com/open?id=0BzmYQVmw4W7nd1o2bFlhTmtLSWM) do apache como a primeira etapa, para que possamos colocar o conteúdo web, que é o arquivo que os diretores passaram para nós.

Uma vez que o download foi concluído, moveremos o conteúdo para o diretório `/var/www/html`. O primeiro passo é instalar o **Apache**:

```
$ sudo apt-get install apache2
```

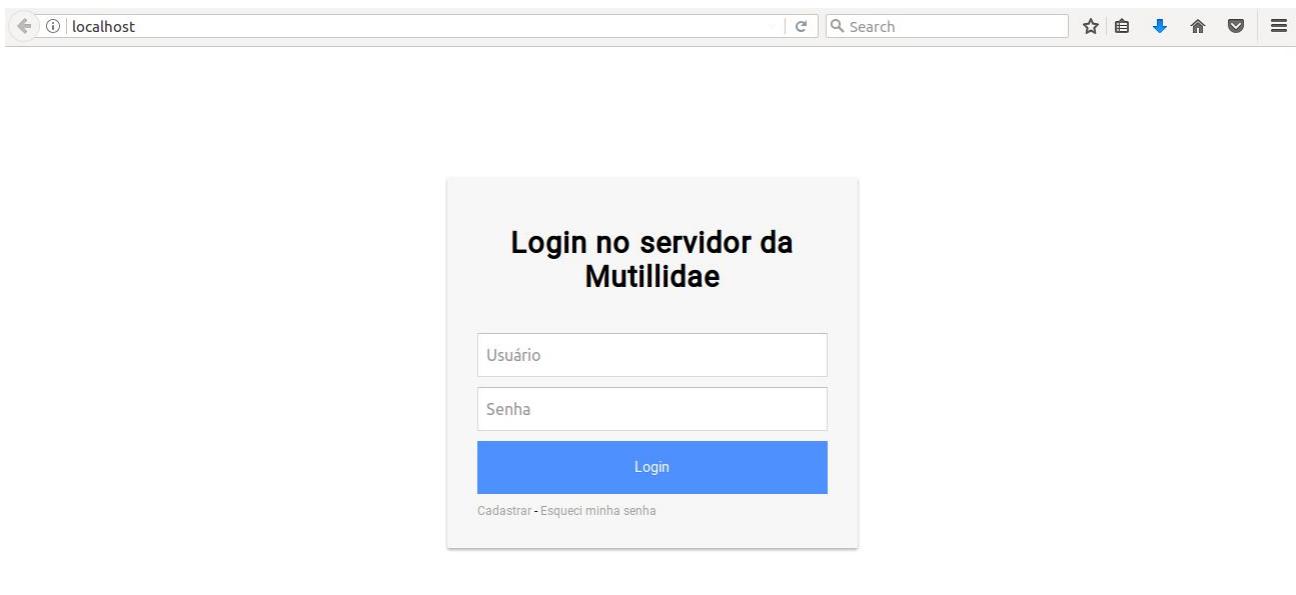
Depois de inserir a senha, será feito o download. Após o término do download do apache e de ter movido o conteúdo para o diretório `/var/www/html`, inicializaremos o servidor do Apache, assim sendo possível ver o conteúdo do serviço interno que os usuários da *Multillidae* irão acessar.

Para inicializar o serviço do servidor Apache, digitamos esse comando:

```
$ sudo service apache2 start
```

Se tudo deu certo, devemos ser capazes de ver o conteúdo que está em `/var/www/html/index.html`, o arquivo que os diretores passaram para nós.

Para acessar esse conteúdo, abrimos uma página no navegador, e digitamos: `localhost/index.html`, ou simplesmente podemos digitar `localhost`, que ele já pegaria o conteúdo do **index.html**. Essa é a tela de login que será acessada pelos usuários da *Multillidae*:



De agora em diante, precisamos nos preocupar em garantir que esse serviço está disponível para os usuários da empresa. O que podemos fazer?

Sempre que acessamos o conteúdo web, fazemos uma **requisição HTTP**. E como retorno, temos um código dizendo se de fato a requisição está ok ou está com algum problema. Tanto é que, quando inspecionamos a página, entramos na aba **Network**, e realizamos uma nova requisição do tipo `GET` para a página que está no servidor Apache, temos o **status HTTP 200**. E caso tudo esteja ok, obtemos o retorno do status 200. Isso quer dizer que se conseguimos monitorar esse status, conseguimos verificar quando esse servidor está ou não funcionando.

Utilizaremos essa estratégia de verificar qual é o status HTTP quando tentamos acessar a URL do `localhost`! Para fazer a monitoração do status de retorno, utilizaremos o `curl`!

Primeiro, temos que verificar se temos o `curl` instalado, basta colocar o comando `curl --version`. Caso ele não esteja, colocamos o comando `sudo apt-get install curl`.

Uma vez feito o download do `curl`, ele nos ajudará a verificar qual é o status HTTP de acesso à URL do `localhost`. Para acessar o conteúdo do `localhost`, colocamos:

```
$ curl http://localhost
```

Depois que tecemos o "Enter", temos todo o conteúdo HTML e CSS do `localhost`, entretanto, somente nos interessa o **status HTTP**, para verificar se está ok ou se não está.

Então, colocamos novamente o mesmo comando no terminal, porém vamos determinar que a saída (*output*) será descartada para o `/dev/null`. Também precisamos imprimir o status HTTP na tela.

```
$ curl --write-out %{http_code} --output /dev/null http://localhost
```

Obtemos o status HTTP 200, e também obtemos algumas estatísticas, mas como não queremos nada além do status, podemos "silenciá-la":

```
$ curl --write-out %{http_code} --silent --output /dev/null http://localhost
```

Assim, somente é mostrado o status! Se esse código funcionou no terminal, então ele deve funcionar no script. Vamos copiar esse comando que acabamos de testar aqui no terminal, e voltar para o diretório `/Scripts` para criar um novo script, que fará a verificação do status do servidor.

```
$ nano monitoracao-servidor.sh
```

A primeira linha do script é o interpretador:

```
#!/bin/bash
```

Agora vamos colar o comando `curl` que foi testado no terminal, e armazenaremos o resultado na variável `resposta_http`:

```
#!/bin/bash
```

```
resposta_http=$(curl --write-out %{http_code} --silent --output /dev/null http://localhost)
```

Vamos testar e ver se o comando ainda está funcionando corretamente, e vamos imprimir o conteúdo da variável `resposta_http`:

```
#!/bin/bash
```

```
resposta_http=$(curl --write-out %{http_code} --silent --output /dev/null http://localhost)  
echo $resposta_http
```

Saímos com "Ctrl + X" e "Y" para salvar, e podemos rodar o script!

```
$ bash monitoracao-servidor.sh
```

O que é nos retornado é o status de 200, que significa que está tudo bem com o servidor Apache. Agora que já completamos essa etapa, precisamos nos preocupar em como conseguiremos fazer a monitoração do servidor para verificar quando ele cair notificar o administrador de sistemas por e-mail, dizendo que houve um problema.