



Para saber mais: Outras restrições

No último vídeo vimos como podemos adicionar restrições a um serviço utilizando o comando:

```
docker service update --constraint-add
```

[COPIAR CÓDIGO](#)

Utilizamos o comando acima para restringir serviços a funcionarem apenas em nós *managers* ou *workers*.

Porém, também podemos impor outros tipos de restrições, como `id` , `hostname` e o próprio `role` . Vamos ver alguns exemplos!

Caso quiséssemos restringir o serviço de `id ci10k3u7q6ti` para funcionar apenas em um nó com `id t76gee19fjs8` , poderíamos utilizar o comando:

```
docker service update --constraint-add node.id==t76gee19fjs8 ci10k3u7q6ti
```

[COPIAR CÓDIGO](#)

Se o objetivo fosse fazer o serviço rodar apenas em nossa `vm4` por exemplo, uma possibilidade seria utilizar:

```
docker service update --constraint-add node.hostname==vm4 ci10k3u7q6ti
```

[COPIAR CÓDIGO](#)

Por fim, podemos também remover restrições criadas utilizando o comando de atualização passando a flag `--constraint-rm`. Para remover as duas restrições anteriores:

```
docker service update --constraint-rm node.id==t76gee19fjs8 ci10l
```

```
docker service update --constraint-rm node.hostname==vm4 ci10k3u7
```

[COPIAR CÓDIGO](#)

Após esse momento, quaisquer novas réplicas criadas para esse serviço poderão ser alocadas sem restrição alguma!