

Instalação do Mesos utilizando Docker

E utilizando o systemd para rodar o serviço

Este tutorial tem como objetivo auxiliar e prestar suporte para a instalação do Apache Mesos e sua utilização como serviço utilizando o systemd.

O Mesos é composto de 2 partes: Mesos-Master, que distribui o processamento, e, Mesos-Slave, que realiza o processamento. Além disso, é necessário também a instalação do ZooKeeper.

Neste *guia* iremos guiar passo a passo, seguindo a ordem em que cada script deverá ser rodado.

Observações

Para rodar os arquivos `.service`, é necessário utilizar o SystemD. CentOS 7 e CoreOS fazem uso dele. E também é necessário o Docker, pois é aonde de fato iremos rodar o Mesos, e não *diretamente no metal*.

Começando

Os arquivos `.service` deverão ficar no diretório `/etc/systemd/system/`, e os passos seguinte estarão todos *acontecendo* nesse diretório, com os arquivos `.service` lá.

Considerações gerais

Todos os Scripts tem uma forma: `[Unit]`, `[Service]` e `[Install]`.

- `[Unit]`:
É aonde é especificado o nome do serviço (no caso do systemd este tem o nome de *unit*)
 - *Description*: é o nome da *unit*;
 - *After*: é a ordem de execução, no caso é para ser interpretado como: *essa unit só vai ser executada depois que esta outra unit já estiver rodando*;
 - *Requires*: É parecido com o *After*, porém ele não segue uma ordem como ele, ele apenas verifica se a unit está funcionando e, caso positivo, já dispara ela.
- `[Service]`:
Aqui é aonde as *coisas acontecem*, aonde você vai demonstrar o passos para a sua unit estar funcionando.
 - *ExecStartPre*: são passos que acontecem antes do serviço de fato acontecer, exemplos disso podem ser limpezas, remoções de arquivos sujos que podem atrapalhar a execução do seu serviço que é de fato alcançado em;
 - *ExecStart*: É aonde você vai rodar o seu serviço de fato, aqui entra no nosso caso o *docker run*, note que ele não está rodando como um daemon, pois o systemd já faz isso;
 - *ExecStop* é o comando que para o serviço que foi especificado em *ExecStart*, no nosso caso, é um *docker stop*;

- *Restart* é um parâmetro para controle de reinicialização caso alguns dos passos de execução falhe para que ele tente novamente ou simplesmente tente uma vez e pare

Execução das Unit

Para despachar as units é bem simples, o comando descrito abaixo roda cada uma delas:

```
# systemctl start zookeeper.service
# systemctl start mesos-master.service
# systemctl start mesos-slave.service
# systemctl start marathon.service
# systemctl start chronos.service
```

Porém, vai ser necessário atualizar os arquivos `.service` para a realidade em questão, alterar IPs, diretórios de volumes, entre outras coisas, para se adaptar ao ambiente que está sendo utilizado.

Zookeeper

Como vamos utilizar apenas uma instância do Zookeeper, este pode ficar do jeito que *está*.

Mesos Master

No script mesos master devemos no atentar para as seguintes variáveis de ambiente (são aquelas que são especificados dentro dos parentêses), e para os volumes, pois devemos colocar as credenciais neles.

1. "MESOS_HOSTNAME": é o nome do Host, nenhum grande impacto, mas serve para diferenciar os hosts; TODO
2. "MESOS_IP": é o IP da máquina que está sendo executado o script; TODO
3. "MESOS_ZK": É um endereço do Zookeeper, formatado do jeito que está no arquivo. Ele serve para registrar o Mesos no Zookeeper, para ser facilmente acessado depois;
4. "MESOS_PORT": Porta padrão que o mesos irá expor para se comunicar;
5. "MESOS_QUORUM": TODO
6. "MESOS_LOG_DIR": diretório para aonde será escrito os logs do Mesos;
7. "MESOS_CREDENTIALS": Diretório o qual estão as credenciais do Mesos;
8. "MESOS_AUTHENTICATE": Condicional para autenticação geral do mesos;
9. "MESOS_AUTHENTICATE_SLAVES": Condicional para autenticação de Slaves no Master do Mesos;
10. "MESOS_CLUSTER": Nome do cluster que está rodando o Mesos.

Mesos Slave

Muito semelhante ao script do Mesos-Master, algumas diferenças tem que ser consideradas:

1. A variável de ambiente **MESOS_MASTER** é passada como um endereço do Zookeeper, pois assim fica mais fácil identificar os IPs de uma rede, centralizando o conhecimento dos IPs;
2. **MESOS_PORT** é uma diferente (no caso 5051) pois estamos executando em uma mesma máquina, caso estejamos executando em máquinas diferentes, não tem nenhum problema ela ser 5050, inclusive, esse é o padrão;
3. O resto das variáveis de ambiente são otimizadas para rodar no Linux sob o Docker.

Marathon

Deverá se atentar apenas para as variáveis **MARATHON_HTTP_PORT** que é a porta que será exposta, para não acontecer nenhum conflito com outra, e o endereço do Mesos-Master, que é descrito em **MARATHON_MASTER**, que também é um endereço do Zookeeper.

Chronos

Além das considerações acima, deverá também ser levado em conta as variáveis que remetem ao email.