Utilizando programas através do Apptainer

O Apptainer é um programa para criação e execução de containers, compatível com o Docker e que permite rodar programas dentro de 'imagens' prontas. Isso evita problemas de compatibilidade entre pacotes do sistema e os requeridos pelo programa.

Sua principal vantagem em relação ao Docker é não precisar de permissões especiais dos usuários, tornando-o mais adequado a nossos servidores de uso compartilhado.

Quase tudo que serve para o Singularity serve para o Apptainer.

Abaixo um guia de como rodar um programa dentro do ambiente (no caso, o R).

Imagens pré-baixadas

Para facilitar, deixaremos algumas imagens baixadas em `/opt/apptainer`.

Para rodar o R com a imagem ml-verse:

```
apptainer exec /opt/apptainer/ml-verse_latest.sif R
```

Rodar o `Rscript` é análogo.

Para o RStudio temos um script que chama a imagem do ml-verse, sendo necessário especificar a porta da interface web do RStudio. É preciso escolher uma porta (qualquer número de 10000 a 40000 que já não esteja ocupado por outro processo no servidor) e acessar o servidor da seguinte forma (no exemplo, foi escolhido o número 12345):

```
ssh -L 12345:localhost:12345 usuario@brucutu.ime.usp.br
```

Então, chamar o script com o mesmo número:

```
rstudio_apptainer.sh 12345
```

O script irá gerar e exibir uma senha de acesso. Basta acessar pelo navegador o endereço http://localhost:12345 e usar a senha para acessar o RStudio.

Pode ser interessante rodar o rstudio_apptainer.sh dentro de uma screen.

Baixando uma imagem

As imagens do Apptainer são ambientes com as dependências necessárias para que o programa funcionar. Para baixar uma imagem podemos usar o subcomando `pull`, como:

```
apptainer pull docker://rocker/ml-verse
```

O exemplo baixa uma imagem do R do Projeto Rocker que já vem com o R dentro de um ambiente Ubuntu com um repositório de pacotes pré-configurado para instalação mais rápida. A imagem deve ser baixada no diretório onde o comando foi executado, com extensão .sif (no exemplo, seria `ml-verse_latest.sif`).

O comando aceita URIs que apontem para imagens que seguem o padrão OCI. Imagens de docker como a do exemplo costumam funcionar.

Executando uma imagem

Para executar a imagem, utilizamos o subcomando `apptainer run`, especificando o nome do arquivo da imagem:

```
# não deve funcionar, pois o container chama o RStudio por padrão e precisará de alguns bind mounts
apptainer run ml-verse_latest.sif
```

Também é possível executar um comando no escopo da imagem:

```
apptainer exec ml-verse_latest.sif R
```

Ao executar o comando, o apptainer irá criar um container baseado na imagem e executar o programa.

Particularidades do ambiente

O apptainer irá automaticamente disponibilizar seu `$HOME` dentro do ambiente, então seus arquivos devem continuar a ser acessíveis normalmente. A maior diferença se dá nas bibliotecas e outras dependências inclusas na imagem. Caso algum pacote dependa de bibliotecas que não estejam inclusas, o ideal é baixar ou criar uma imagem mais completa.

Particularidades dos servidores do IME

Além do `$HOME`, o apptainer já disponibiliza o `/tmp` e o `/var/tmp` (entre outros diretórios). Nas brucutus também configuramos acesso ao `/var/fasttmp`, que é sempre um SSD.

Referências

https://apptainer.org/docs/user/latest/

From:
https://wiki.ime.usp.br/ - **Wiki da Rede IME**

Permanent link:
https://wiki.ime.usp.br/tutoriais:apptainer?rev=1710533000

Last update: **2024-03-15 17:03**