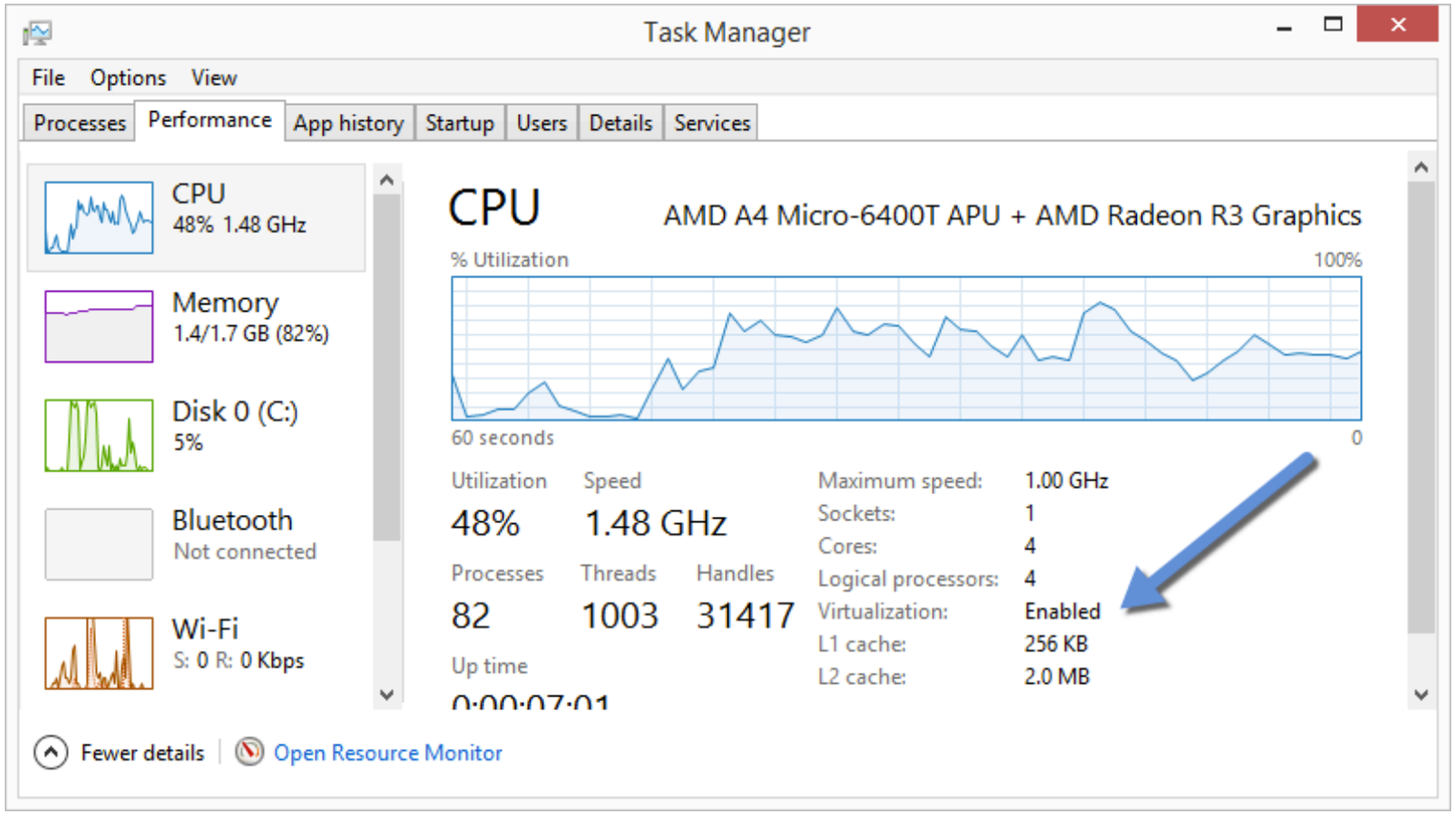
Introdução ao Docker
Comandos para Exercício

Residência de Software
Iniciação à Tecnologia da Informação

Instalação a partir do Windows

- Docker Toolbox
- versões 64bit do Windows e versões superiores ao Windows 7
- Necessário habilitar suporte de virtualização



Instalação Docker Desktop

- Baixar instalador do Windows no site

- https://www.docker.com/products/docker-desktop

- https://hub.docker.com/?overlay=onboarding

- Proceder a instalação

- Usar o software Windows PowerShell , que executa o processo para começar a utilizar o Docker

- Execute o seguinte comando para teste: docker container run hello-world

Instalação a partir do Linux

- Instalar Docker no Ubuntu (Não é um dos repositórios oficiais)

- Se utilizar VPS, acesse-o primeiramente

- Atualizar o sistema para receber o Docker sem conflitos
  - $ sudo apt update
  - $ sudo apt upgrade

- Instalar pacotes
  - $ sudo apt-get install curl apt-transport-https ca-certificates software-properties-common

Instalação a partir do Linux

- Adicionar os Repositórios do Docker:

- Adiciona chave GPG
- $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

- Adiciona repositório
- $ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $ (lsb_release -cs) stable”

- Atualiza informações do repositório
- $ sudo apt update

- Instala o Docker
- $ sudo apt install docker-ce

- Verifica o status da instalação
- $ sudo systemctl status docker

Instalação a partir do Mac Os

- Pode ser instalado por download seguindo os mesmos passos do Windows

- Ou pode ser realizada através de um único grande pacote chamado Docker for Mac.

- Via brew cask com o comando
    - $ brew cask install docker

- Para efetuar a configuração inicial, executar o aplicativo Docker

- Será solicitado seu usuário e senha para liberar a instalação dos softwares.

Exemplo de uso do Docker (1)

- Clonar um exemplo para criar um container, a partir de um repositório do Docker
  - $ git clone https://github.com/docker/doodle.git

- Criar uma imagem do Docker usando o Dockfile
  - No windows
    - $ cd doodle\cheers2019 ; docker build -t <usuário>/cheers2019 .
  - Em sistemas Unix
    - $ cd doodle/cheers2019 && docker build -t <usuário>/cheers2019 .

Exemplo de uso do Docker (2)

- Rodar o container
    - $ docker run -it --rm <usuário>/cheers2019

- Enviar imagem ao Docker Hub
    - No Windows
        - $ docker login ; docker push <usuário>/cheers2019
    - Em sistemas Unix
        - $ docker login && docker push <usuário>/cheers2019

Atividade

- Proceder os mesmos passos para clonar, criar imagem, rodar e enviar para o hub, mas alterando o termo cheers2019 por:

- summer2019

- birthday2019

Comandos Básicos Executando um Container

- Listar as imagens que o Docker host tem localmente
  - docker image list

- Atualizar a imagem
  - docker image pull <imagem>

- Inspecionar a imagem que acabou de atualizar
  - docker image inspect <imagem>

- iniciar o container:
  - $ docker container run -it --rm --name <container> <imagem> bash

Comandos Básicos
Mapeamento de Volumes

- Para mapeamento de volume, especificar origem do dado no host e onde deve ser montado dentro do container
  - $ docker container run -it --rm -v "<host>:<container>" <imagem>

- Para mapear portas, saber qual porta será mapeada no host e qual vai receber essa conexão dentro do container
  - $ docker container run -it --rm -p "<host>:<container>" <imagem>

- Exemplo: porta 80 do host para uma porta 8080 dentro do container
  - $ docker container run -it --rm -p 80:8080 <imagem>

Comandos Básicos Gerenciamento de Recursos

- É possível especificar limites de utilização dos recursos do container

- Limitar o uso de memória RAM utilizada por esse container
  - $ docker container run -it --rm -m <quant_memória> <imagem>
  - Exemplo: $ docker container run -it --rm -m 512M <imagem>

- Especificar prioridade de uso do container (peso vai de 1 a 1024)
  - $ docker container run -it --rm -c <peso> <imagem>

Comandos Básicos Mapeamento de Volumes

- Visualizar lista de containers de um Docker host
  - $ docker container ls
  - $ docker container ls <parâmetros>

Parâmetro	Explicação
-a	Lista todos os containers, inclusive os desligados
-l	Lista os últimos containers, inclusive os desligados
-n	Lista os últimos N containers, inclusive os desligados
-q	Lista apenas os ids dos containers, ótimo para utilização em scripts

Comandos Básicos Gerenciamento de Containers

- Desligar o container
  - docker container stop <nome_container_ou ID>

- Reiniciar o container que foi desligado, não iniciando um novo
  - docker container start <nome_container_ou ID>

Criando Imagens
Commit

- É possível criar imagens com base no status atual de um container
- Exemplo:
  - Cria-se um container:
    - $ docker container run -it --name <containercriado> ubuntu:16.04 bash
  - Instala nginx
    - $ apt-get update apt-get install nginx -y exit
  - Paralisa o container
    - $ docker container stop <containercriado>
  - Efetua commit em uma imagem
    - $ docker container commit <containercriado> meuubuntu:nginx

Criando Imagens
Commit

- **containercriado** - nome do container criado e modificado
- **meuubuntu:nginx** - imagem resultante do commit
- **meuubuntu:nginx** - imagem que guarda estado do containercriado

- Testar a nova imagem:
- Criar um container a partir da imagem e ver se o nginx está instalado:
- $ docker container run -it --rm meuubuntu:nginx dpkg -l nginx
- Para validar, executar mesmo comando na imagem oficial do ubuntu
- $ docker container run -it --rm ubuntu:16.04 dpkg -l nginx

- commit não é a melhor opção para criar imagens
- processo de modificação é completamente manual
- apresenta certa dificuldade para rastrear mudanças

Criando Imagens Dockerfile

- Um arquivo Dockerfile, representa a diferença entre uma imagem (base), e a imagem que se deseja criar.

- Exemplo:
  - Criar arquivo para teste futuro:
    - touch arquivo_teste
  - Criar arquivo chamado Dockerfile e dentro dele o conteúdo:
    - FROM ubuntu:16.04
    - RUN apt-get update && apt-get install nginx -y
    - COPY arquivo_teste /tmp/arquivo_teste
    - CMD bash

Criando Imagens
Dockerfile

- FROM informa qual imagem será a base

- RUN informa quais comandos serão executados para efetuar as mudanças necessárias na infraestrutura do sistema.

- COPY copia arquivos da estação onde está executando a construção para dentro da imagem.

- CMD para informa qual comando será executado por padrão

Criando Imagens Dockerfile

- Construir imagem por Dockerfile
  - $ docker image build -t meuubuntu:nginx_auto .

- Arquivo Dockerfile é uma sequência de instruções lidas do topo à base. Uma linha executada por vez

- Se alguma instrução depender de outra instrução, essa dependência deve ser descrita mais acima no documento

- Cada instrução do arquivo é armazenado em cache local