**O que é Docker?**

* Não é um sistema de virtualização tradicional;
* É engine de administração de containers;
* Utiliza os serviços do LXC (Linux Containers);
* Open Source e escrito em Go;
* Sistema de virtualização baseado em software (SO);
* Host e Container compartilham o kernel;
* Empacota software com vários níveis de isolamento.

**Por que não uma VM?**

* Mantém as qualidades e consome muito menos recursos;
* Não precisa de um sistema operacional completo.

**O que são containers?**

* Segregação de processos no mesmo Kernel (Isolamento);
* Sistemas de arquivos criados a partir de uma imagem;
* Ambientes leve e portáteis no qual aplicações são executadas;
* Encapsula todos os binários e bibliotecas necessárias para execução de uma App;
* Algo entre um chroot e uma VM.

**O que são imagens Docker?**

* Modelo de sistema de arquivo somente-leitura usado para criar containers;
* Imagens são criadas através de um processo chamado build;
* São armazenadas em repositórios no Registry;
* São compostas por uma ou mais camadas (layers);
* Uma camada representa uma ou mais mudanças no sistema de arquivo;
* Uma camada é também chamada de imagem intermediária;
* A junção dessas camadas forma a imagem;
* A apenas a última camada pode ser alterada quando o container for iniciado;
* AUFS. (Advanced multi-layered unification filesystem) é muito usado;
* O grande objetivo dessa estratégia de dividir uma imagem em camadas é o reuso;
* É possível compor imagens a partir de camadas de outras imagens.

**Imagem vs Container**

* Imagem Docker é como se fosse um modelo e o container é a realização desse modelo;
* Container é o processo e a imagem é o modelo em que o container se baseia.

**Arquitetura**

