



CAPTURING REQUIREMENTS FOR CONTAINERS
LINK TO MY DOCKERHUB REGISTRY
Hit Your Right Arrow To Begin!

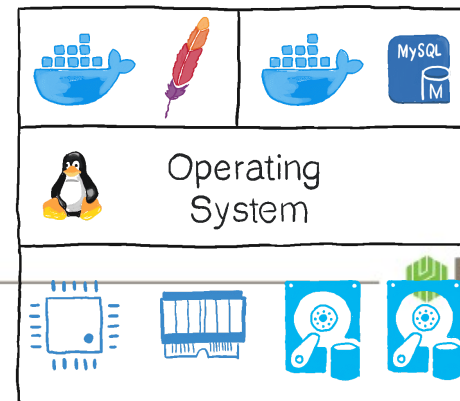
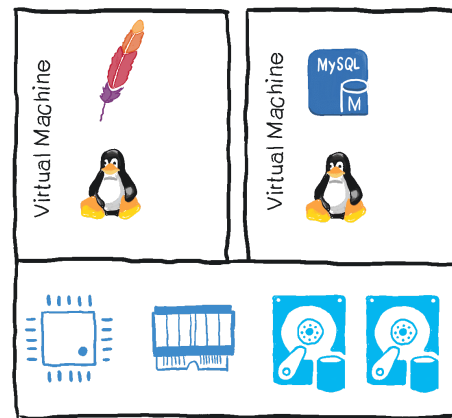
AGENDA

1. What are Containers and Why use them?
2. Orchestration with Kubernetes
3. MONOLITH VERSUS MICROSERVICES
4. MICROSERVICES: ARCHITECTURE TO SUPPORT AGILE
5. HOW ARE REQUIREMENTS NEW WITH CONTAINERIZATION
6. SECURITY REQUIREMENT WITH CONTAINERS
7. REFERENCES



WHAT ARE CONTAINERS ?

Containers offer a logical packaging mechanism in which applications can be abstracted from the environment in which they actually run.



Huntington

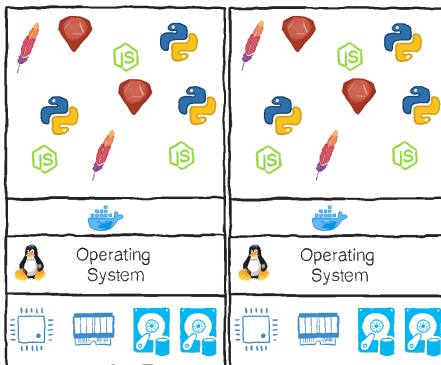
...SO WHY ARE WE TALKING ABOUT IT ?

We can get rid of - "I don't know why it's not working for you ?"

GREAT FOR SERVICE BASED ARCHITECTURE
Containers offers variety of Performance benefits



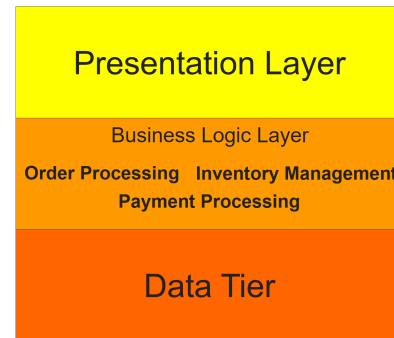
Orchestration



- Running containers across many different machines
- Scaling up or down by adding or removing containers when demand changes
- Distributing load between the containers
- Distributing load between the containers
- Launching new containers on different machines if something fails

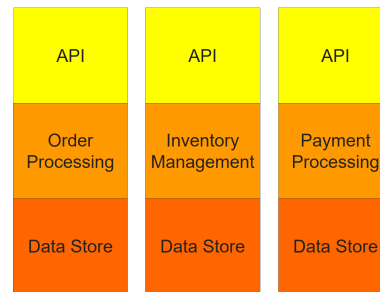


CHALLENGES WITH MONOLITHIC ARCHITECTURE



- Application is too large and complex.
- Impact of a change is usually not very well understood which leads to do extensive manual testing.
- Another problem with monolithic applications is reliability. Bug in any module (e.g. memory leak) can potentially bring down the entire process.

ENTER MICROSERVICES



- Split your application into a set of smaller, interconnected services instead of building a single monolithic application.
- These services are built around business capabilities and are independently deployable .
- Services communicate with each other by using well-defined APIs. Internal implementation of each service are hidden from other services.



AGILE INTEGRATION

- 1) IT WANTS TO BE MORE RESPONSIVE TO THE CHANGES IN THE OVERALL TECH INDUSTRY; THE DRIVING NEED TO MOVE FASTER.**
- 2) BUSINESS COMPONENTS OR SERVICES TOGETHER QUICKLY AND RESPOND TO MARKET DEMANDS**
- 3) USE OF SMALLER TEAMS AND AN INCREMENTAL APPROACH**
- 4) INTEGRATION UP FRONT**



HOW ARE REQUIREMENTS NEW WITH CONTAINERIZATION?

- 1) ORGANIZATIONS ARE ORGANIZING THEIR DEVELOPMENT TEAMS DIFFERENTLY AND THEY NEED TO GET BUSINESS ELEMENTS INVOLVED.**
- 2) MICROSERVICES ARE ORGANIZED AROUND BUSINESS CAPABILITIES**
- 3) APIS IS NOW CONSIDERED THE THIRD PILLAR FOR AGILE.**
- 4) SCALIBILITY AND PERFORMANCE IS A LOT DIFFERENT WITH CONTAINERS**



CONTAINERS ARE HERE , NOW HOW DO WE SECURE THEM?

- 1) INTEGRATE STATIC APPLICATION SECURITY TESTING INTO THE BUILD PROCESS .
- 2) ROBUST, CENTRALIZED AUTHENTICATION AND AUTHORIZATION POLICIES FOR ACCESS.
- 3) ENCRYPT ALL DATA AT REST. FULL DISC ENCRYPTION
- 4) MAKE SURE THAT YOUR PLATFORM IS MOSTLY "ATOMIC".



DESIGNED BY - TARANJEET KAUR

