



# Spring Boot Microservices

---

Beginner to Guru

Deployment Design



## Containerized Deployment

- Microservices are well suited for containerized deployment
- Typically are compute only, and do not persist data in instance
- Scalability and reliability achieved with multiple instances
- Deployments managed via container orchestration
  - Docker Swarm is among the simplest - uses extensions in Docker Compose
- Other more robust solutions include Kubernetes, OpenShift, Mesos, AWS ECS
  - This is a large and evolving area





## Containerized Deployment

- Databases persisting data typically are poor candidates for containerized deployments
  - Becomes a problem in managing disk storage
  - While it can be done, typically not the optimal solution
  - Often you will see dedicated VMs or physical servers for databases
    - Nothing faster than physical servers
- This extends to any database like application
  - JMS or other message brokers, Elasticsearch, Zipkin, etc





## Deployment Goals

- Use Digital Ocean to create a realistic deployment
  - Be pragmatic and highlight where a actual production deployment would differ
- Use Digital Ocean Managed MySQL databases
  - Setup 3 - one per microservice
  - Larger organizations will have a database administration team
- Setup a dedicated JMS broker
  - High volume production would use a cluster





## Deployment Goals

- Setup Dedicated Eureka Server
  - Production would have a cluster for high availability
- Setup Dedicated ElasticSearch Server
  - Production would have cluster for high availability and scalability
- Setup Dedicated Zipkin Server
  - Production would use a Cassandra or ElasticSearch data store







## Deployment Goals

- Setup Dedicated Configuration Server
  - Production would use a cluster for high availability
- Deploy Spring Boot Services to 3 node Swarm Cluster
  - Gateway, Beer Service, Inventory Service, Inventory Failover Service, Order Service
  - Use Spring Cloud Config with new profile for cloud deployment
- Filebeat
  - Deploy per node, use 'extra\_hosts' to config Elasticsearch Server





## Summary

- Deployment needs 12 different servers (including 3 MySQL instances)
  - 1 GB RAM / 1 vCPU
- For simplicity we will use Docker deployments
- VMs for services - 6 VMs with 4GB
- VMs for Swarm Cluster - 3 VMs with 8GB

