Pattern: Multiple service instances per host

Context

You have applied the Microservice architecture pattern (/patterns/microservices.html) and architected your system as a set of services. Each service is deployed as a set of service instances for throughput and availability.

Problem

How are services packaged and deployed?

Forces

- Services are written using a variety of languages, frameworks, and framework versions
- Each service consists of multiple service instances for throughput and availability
- Service must be independently deployable and scalable
- Service instances need to be isolated from one another
- You need to be able to quickly build and deploy a service
- You need to be able to constrain the resources (CPU and memory) consumed by a service
- You need to monitor the behavior of each service instance
- You want deployment to reliable
- You must deploy the application as cost-effectively as possible

Solution

Run multiple instances of different services on a host (Physical or Virtual machine).

There are various ways of deploying a service instance on a shared host including:

- Deploy each service instance as a JVM process. For example, a Tomcat or Jetty instances per service instance.
- Deploy multiple service instances in the same JVM. For example, as web applications or OSGI bundles.

Examples

Resulting context

The benefits of this pattern include:

- More efficient resource utilization than the Service Instance per host pattern (single-service-per-host.html)

The drawbacks of this approach include:

- Risk of conflicting resource requirements
- Risk of conflicting dependency versions
- Difficult to limit the resources consumed by a service instance
- If multiple services instances are deployed in the same process then its difficult to monitor the resource consumption of each service instance. Its also impossible to isolate each instance

Related patterns

- The Single Service Instance per Host pattern (single-service-per-host.html) is an alternative solution.
- The Serverless deployment pattern (/patterns/deployment/serverless-deployment.html) is an alternative solution.