Desenvolvimento de Aplicações com Arquitetura Baseada em Microservices

Prof. Vinicius Cardoso Garcia
vcg@cin.ufpe.br :: @vinicius3w :: assertlab.com

[IF1007] - Tópicos Avançados em SI 4
https://bit.ly/vcg-microservices

Licença do material

The Deployment Pipeline

Overall Architecture

A distributed system is one in which the failure of a computer you didn’t even know existed can render you own computer unusable.
— Leslie Lamport

Introduction

- What are the structural implications of the DevOps practices?
  - both the overall structure of the system and techniques that should be used in the system's elements
- DevOps achieves its goals partially by replacing explicit coordination with implicit and often less coordination
  - the architecture of the system being developed acts as the implicit coordination mechanism

Do DevOps Practices Require Architectural Change?

- If you must re-architect your systems in order to take advantage of DevOps, a legitimate question is "Is it worth it?"

- Some DevOps practices are independent of architecture,
- whereas in order to get the full benefit of others, architectural refactoring may be necessary

AssertLab
Advanced Software and Systems Engineering Research Technologies

Recall the 5 categories of DevOps practices

1. Treat Ops as first-class citizens from the point of view of requirements
   - Operations have a set of requirements that pertain to logging and monitoring
2. Make Dev more responsible for relevant incident handling
3. Enforce the deployment process used by all, including Dev and Ops personnel
   - Ensure a higher quality, avoids errors and the resulting misconfiguration
4. Use continuous deployment
   - Shorten the time between a developer committing code to a repository and the code being deployed
5. Develop infrastructure code, such as deployment scripts, with the same set of practices as application code

Overall Architecture Structure

- Warm up
  - a module is a code unit with coherent functionality
  - a component is an executable unit

- Development teams using DevOps processes are usually small and should have limited inter-team coordination
  - integration and acceptance tests are mandatory

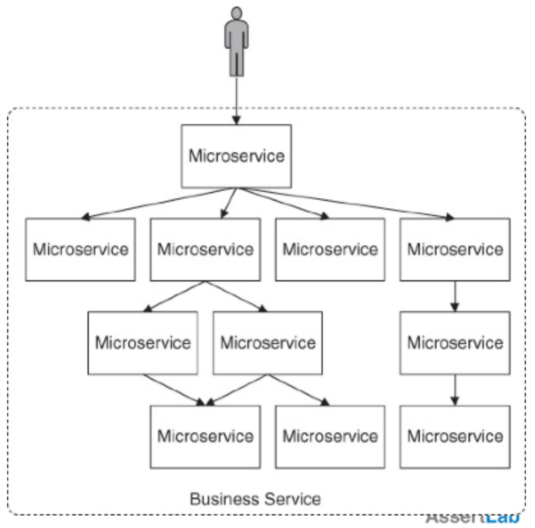
Overall Architecture Structure

- An organization can introduce continuous deployment without major architectural modifications
  - Deploying without the necessity of explicit coordination with other teams reduces the time required to place a component into production.
  - Allowing for different versions of the same service to be simultaneously in production leads to different team members deploying without coordination with other members of their team.
  - Rolling back a deployment in the event of errors allows for various forms of live testing

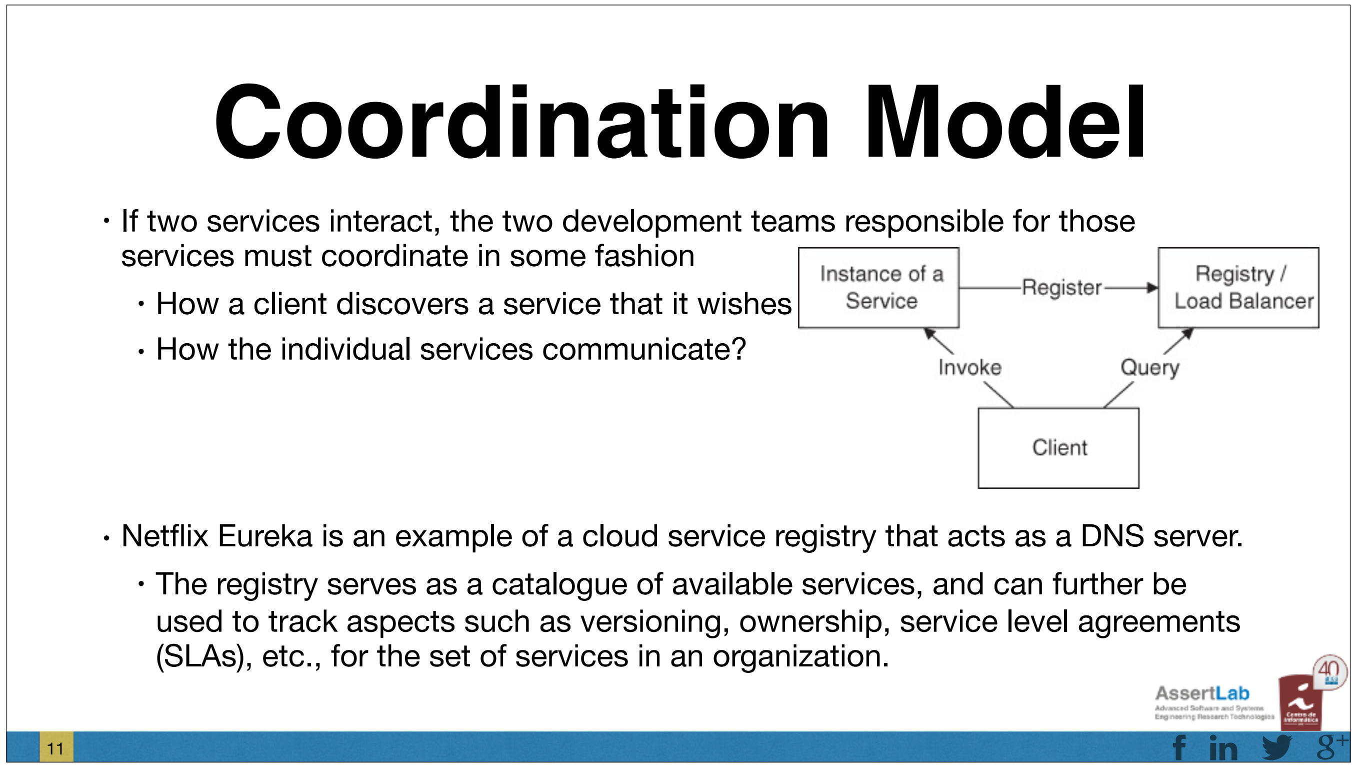
- Microservice architecture is an architectural style that satisfies these requirements

Microservice Architecture

“A microservice architecture consists of a collection of services where each service provides a small amount of functionality and the total functionality of the system is derived from composing multiple services"

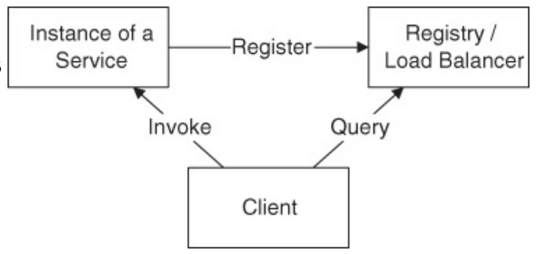


A user interacts with a single consumer-facing service. This service, in turn, utilizes a collection of other services. We use the terminology service to refer to a component that provides a service and client to refer to a component that requests a service. A single component can be a client in one interaction and a service in another. In a system such as LinkedIn, the service depth may reach as much as 70 for a single user request.



Coordination Model

- If two services interact, the two development teams responsible for those services must coordinate in some fashion
  - How a client discovers a service that it wishes
  - How the individual services communicate?



- Netflix Eureka is an example of a cloud service registry that acts as a DNS server.
  - The registry serves as a catalogue of available services, and can further be used to track aspects such as versioning, ownership, service level agreements (SLAs), etc., for the set of services in an organization.

“The service registers with a registry. The registration includes a name for the service as well as information on how to invoke it, for example, an endpoint location as a URL or an IP address. A client can retrieve the information about the service from the registry and invoke the service using this information. If the registry provides IP addresses, it acts as a local DNS server—local, because typically, the registry is not open to the general Internet but is within the environment of the application.

Management of Resources

- Two types of resource management decisions can be made globally and incorporated in the architecture
  - provisioning/deprovisioning VMs
  - managing variation in demand.

Provisioning & Deprovisioning VMs

- New VMs can be created in response to client demand or to failure
  - If the instances are stateless, a new instance can be placed into service as soon as it is provisioned
  - Similarly, if no state is kept in an instance, deprovisioning becomes relatively painless
- An additional advantage of a stateless service is that messages can be routed to any instance of that service, which facilitates load sharing among the instances.

Provisioning & Deprovisioning VMs

- This leads to a global decision to maintain state external to a service instance (see lecture #3)
-