1. Converting to Microservices

A case-study in devops transformation

Summary

As part of your homework, please read the abstract below and propose a plan of action for an engagement with this client. Many of our pre-sales opportunities start like this and based on our response to a problem statement like this one, we move on to a more detailed assessment and call. Your goal is to use what you know from the abstract and demonstrate that you understand the problem as well as some of the details that will likely be part of any solution. It should be ready to send to a customer.

Abstract

Z Widgets Ltd is considering proposals from several companies to safely move their application from a C# monolith on Windows with IIS on AWS to a new set of NodeJS microservices deployed in docker containers. The expectation is that they will stick to AWS, but that is open for discussion if there's a better value elsewhere. There are three motivations for this move: 1. The difficulty of hiring new C# engineers. 2. The internal team wants to develop nodejs. 3. The existing stack has reached the limit of vertical scaling tied to a single Microsoft SQL Server. The current architecture uses the largest MS SQL RDS instance available with 12 IIS instance for delivering the existing REST web services through a set of related ELBs. There are also five IIS instances that serve the website HTML. There are also a host of other services involved with the backend processing, but not directly involved with the web stack. All of these are being managed manually by a team of four IT engineers who are willing and eager to help with the transition, but they can't figure out how to do it without significant disruption to the running services and consequently, the customers.

2. Using a vagrantfile which executes any configuration management framework (e.g. chef, puppet, ansible or salt) or docker, automatically stand up a data driven website (of any stack) from scratch.

3. In any language, write a script that:
    a. Takes an ip address and a list of ports as standard input.
    b. Check if each of the ports is active.
    c. If the port is down, send a command to turn the service on via ssh.
    d. Write the results to a log.
    e. extra credit: include a vagrantfile to run the script against

4. Create a vagrant box that runs hadoop and monitoring tools watching it.