Infrastructure as Code

March 21, 2022

Brae Webb

Presented for the Software Architecture course
at the University of Queensland

THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA

Infrastructure as Code   Software Architecture

March 21, 2022   Brae Webb

1   Introduction

Configuration management can be one of the more challenging aspects of software development. In sophisticated architectures there are roughly two layers of configuration management; machine configuration and stack configuration.

Machine configuration encompasses software dependencies, operating system configuration, environment variables, and every other aspect of a machines environment.

Stack configuration encompasses the configuration of an architectures infrastructure resources. Infrastructure includes compute, storage, and networking resources.

In a sense, machine configuration is a subset of the stack configuration, however, stack configuration tend to focus on a higher level of abstraction. In this course, our focus when looking at Infrastructure as Code (IaC) is *stack configuration*. We rely on containerization tools, such as docker, to fill the hole left by the lack of treatment of machine configuration.

2   Brief History

In the 'Iron Age' of computing, installing a new server was rate limited by how quickly you could shop for and order a new physical machine. Once the machine arrived at your company, some number of weeks after the purchase, someone was tasked with setting it up, configuring it to talk to other machines, and installing the all software it needed. Compared to the weeks it takes to acquire the machine, a day of configuration is not so bad. Furthermore, because so much physical effort was needed to provision a new machine, a single developer would only be responsible for a few machines.

With virtualization one physical machine can be the home of numerous virtual machines. Each one of these machines requires configuration. Now, with new machines available within minutes and no physical labour involved, spending a day of configuring is out of the question — introducing: *infrastructure code*.
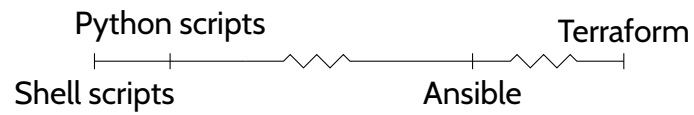
> **Definition 1. Infrastructure Code**
>
> Code that provisions and manages infrastructure resources.

> **Definition 2. Infrastructure Resources**
>
> Compute resources, networking resources, and storage resources.

Infrastructure code arose to ease the burden of the increased complexity where each developer configured and maintained many more machines. Infrastructure code can often be simple. A shell script which installs dependencies is infrastructure code. There's an assortment of infrastructure code out in the world today, ranging from simple shell scripts up to more sophisticated tools such as Ansible and Terraform.

Python scripts ⋯⋯⋯⋯⋯⋯⋯⋯ Terraform
Shell scripts ⋯⋯⋯⋯⋯⋯⋯⋯ Ansible

3 Infrastructure as Code

> **Definition 3. Infrastructure as Code**
> Following the same good coding practices to manage Infrastructure Code as standard code.

3.1 Good Coding Practices

- Everything as Code
- Version Control
- Automation
- Code Reuse
- Testing

References