

SUSEcondigital 

Using Test-Driven Development To Boost Your Infrastructure-as-Code Strategy

DEV-1347

Sean Marlow

Public Cloud Developer

SUSE

Agenda

1. Overview of Infrastructure-as-code (IaC) and Configuration Management
2. Overview of Test-Driven Development (TDD)
3. Walk through an example
4. Closing thoughts



Overview of IaC and Configuration Management



Configuration Management

is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life


~ *Wikipedia*



Salt

is Python-based, open-source software for event-driven IT automation, remote task execution, and configuration management

~ *Wikipedia*



Infrastructure-as-code (IaC)

is the process of managing and provisioning computer data centers through machine-readable definition files

~Wikipedia



Terraform

is an open-source IaC software tool created by HashiCorp which enables users to define and provision a datacenter infrastructure

~ *Wikipedia*



Advantages Of IaC And Configuration Management

- Consistent systems
- Repeatable processes
- Easy to change
- History is visible and traceable



Challenges With IaC And Configuration Management

- Error replication
- Configuration drift
- Erosion
- Fear of automation



Overview of TDD



Test Driven Development

is a software development process where requirements are turned into very specific test cases, then the software is improved so that the tests pass

~ *Wikipedia*



What Do You Gain By Driving IaC With TDD?

- Drives modular and flexible configuration
- Fast feedback
- Prevent scope creep
- Provides an extra layer of documentation
- Drive smaller changes more frequently
- Peace of mind



Pytest

is a mature full-featured Python testing tool



Testinfra

is a framework for writing unit tests in Python to test actual state of your servers configured by management tools like Salt, Ansible, Puppet, Chef



Code Examples



App Requirements

- Python 3 (Flask)
- Apache2 + mod_wsgi
- openSUSE Leap 15.1
- EC2



API

- GET All: /pancakes/
- GET Type: /pancakes/banana
- Add Type (POST): /pancakes/
- DELETE Type: /pancakes/fake



Example Code



Closing Thoughts

Key Takeaways

1. You can test infrastructure-as-code
2. Automated testing of IaC and configuration management is worthwhile

Links

Example Flask App:

<https://github.com/smarlowucf/glowing-pancake>

Presentation:

[**https://smarlowucf.github.io/tdd-iac.html**](https://smarlowucf.github.io/tdd-iac.html)



Thank You!

The background is a solid green color with a white grid pattern that forms wavy, organic shapes. The grid lines are thin and create a mesh-like texture. The overall aesthetic is clean and modern.

SUSEcon digital '20

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE, LLC, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.