Test Driving Your Infrastructure

...

Things We Will Cover

- Standing up an environment using terraform
- Verifying it's configuration using RSpec
- Verifying behaviors of our puppet modules using Test Kitchen and Serverspec
- Verifying our applied configurations using Serverspec against the live environment

Levels of Testing

- Infrastructure / Environment configurations such as proper setup of auto scaling groups, launch configurations, and instances.
- Puppet module behaviors
- Actual implementation of configurations in environments

Environment Setup & Testing

- Terraform
- RSpec
- AWS SDK for Ruby

Puppet Modules

- Test Kitchen using Docker or Vagrant
- Verify behaviors of Puppet modules as we develop them using Serverspec

Applied Configuration

- Serverspec to verify the specific configurations we expect to be applied for each environment.

Let’s Talk About Continuous Integration

Continuous Integration

Jenkins jobs are managed as code using the DSL Plugin. Updates to jobs are done by opening a pull request, and Jenkins server is “just another puppet module”

Continuous Integration

When new environment configuration (Terraform) is applied we run integration tests to ensure systems are configured properly, and all expected services are running.

Continuous Integration

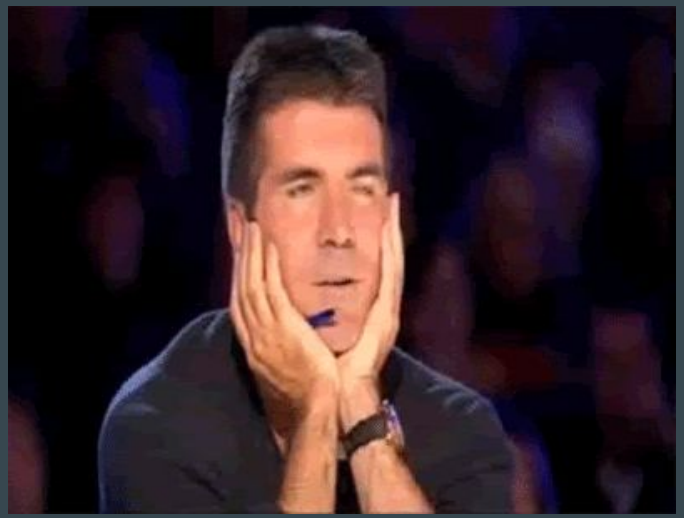
When code is committed to a puppet module we run Test Kitchen tests using an EC2 driver to quickly discover things that only “work on my machine”

Continuous Integration

Teams who work closely with key applications and components have been developing acceptance tests focused on verifying key application components are operating as expected.

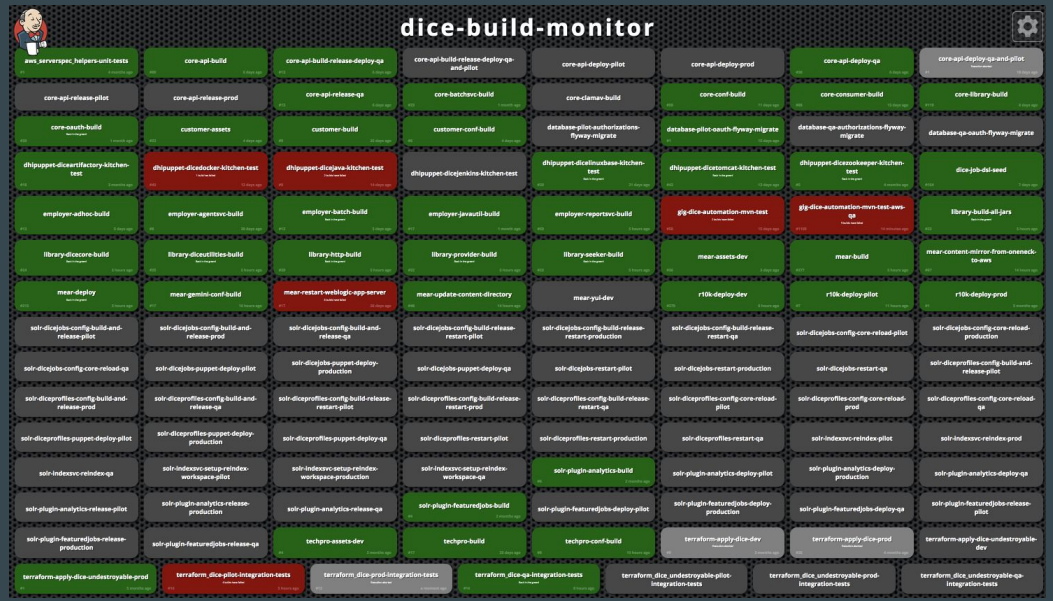
Continuous Integration

Applications are built and deployed to artifactory using Jenkins. Puppet configuration can then be updated to deploy newly built applications. Integration and acceptance tests will verify things are operating as expected after any configuration updates have been applied.



Continuous Integration

All of this is reflected on our build monitor which is visible to anyone in our team space



That sounds like a lot...let's talk benefits

...

Discover Problems Quickly

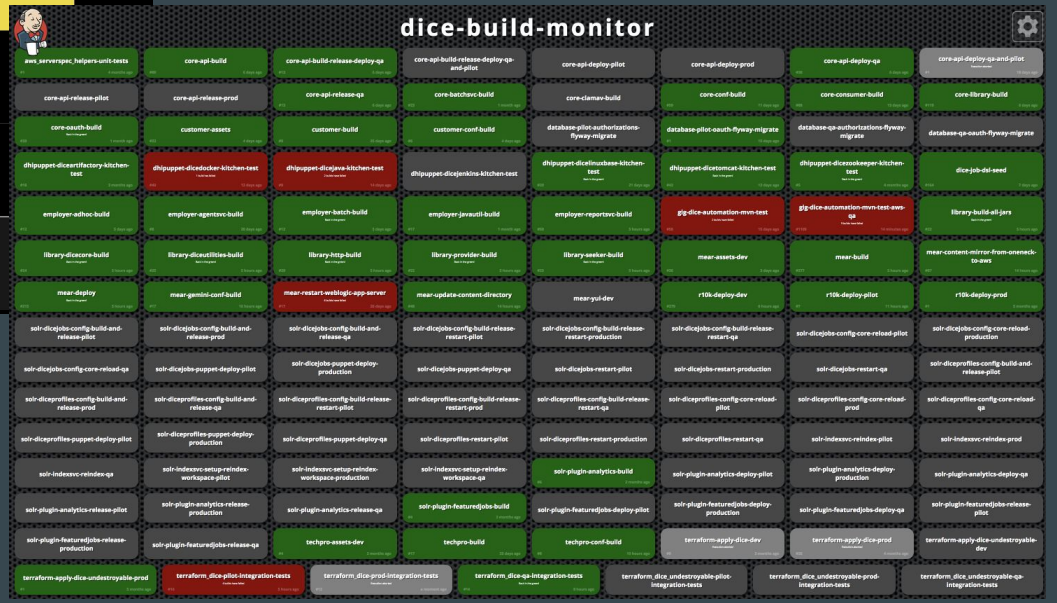
This is a new to us...we experience many failures.

Automation is Reproducible

Since everything is automated we're able to reproduce those failures and resolve them more easily.

Visibility + Reproducibility = Stakeholder Confidence

0 Failed
3 Unreported
26 Pending
0 Changed
0 Unchanged
29 Total



Questions?

Examples and slides: https://github.com/jessiepuls/2017-08-aws-meetup

Feel free to contact me: @jessiepuls