CloudSkills.io

Week 3: Infrastructure as Code

Luke Orellana

@LukeOrellana

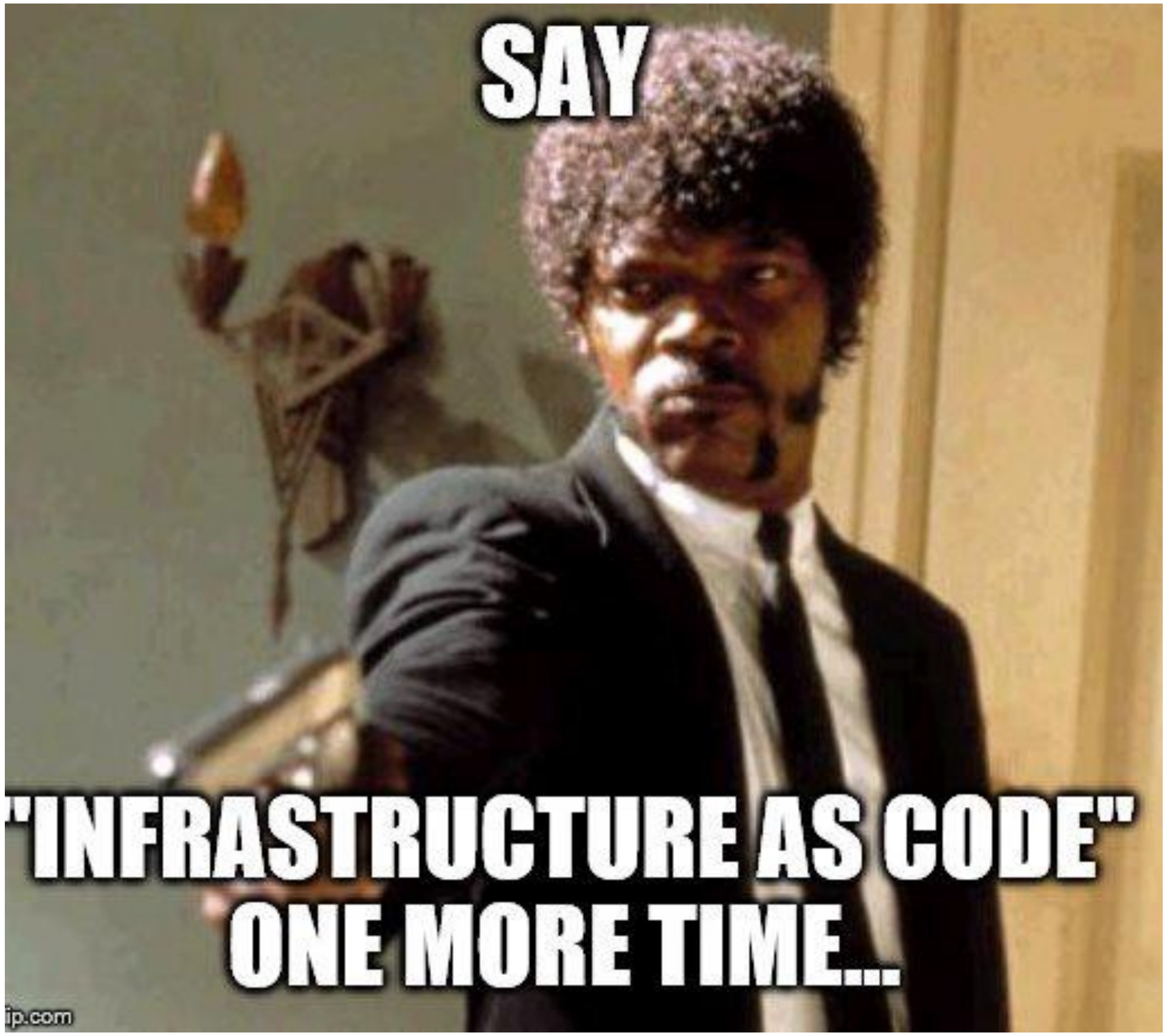
linkedin.com/in/luke-orellana

Course Outline

- **Lecture**: Theory of Infrastructure as Code
- **Projects**:
  - Deploy VM with ARM in Azure
  - Build a Terraform Module to deploy EC2 in AWS
  - Write a Test for the Terraform Module using Terratest
  - Intro to Azure Bicep

What is Infrastructure as Code?

- Defining your infrastructure with code and managing it like a software developer
- DevOps Culture: IaC supports the effort to bridge gaps and improve collaboration



A New Way of Thinking

The game has changed. What's Different?

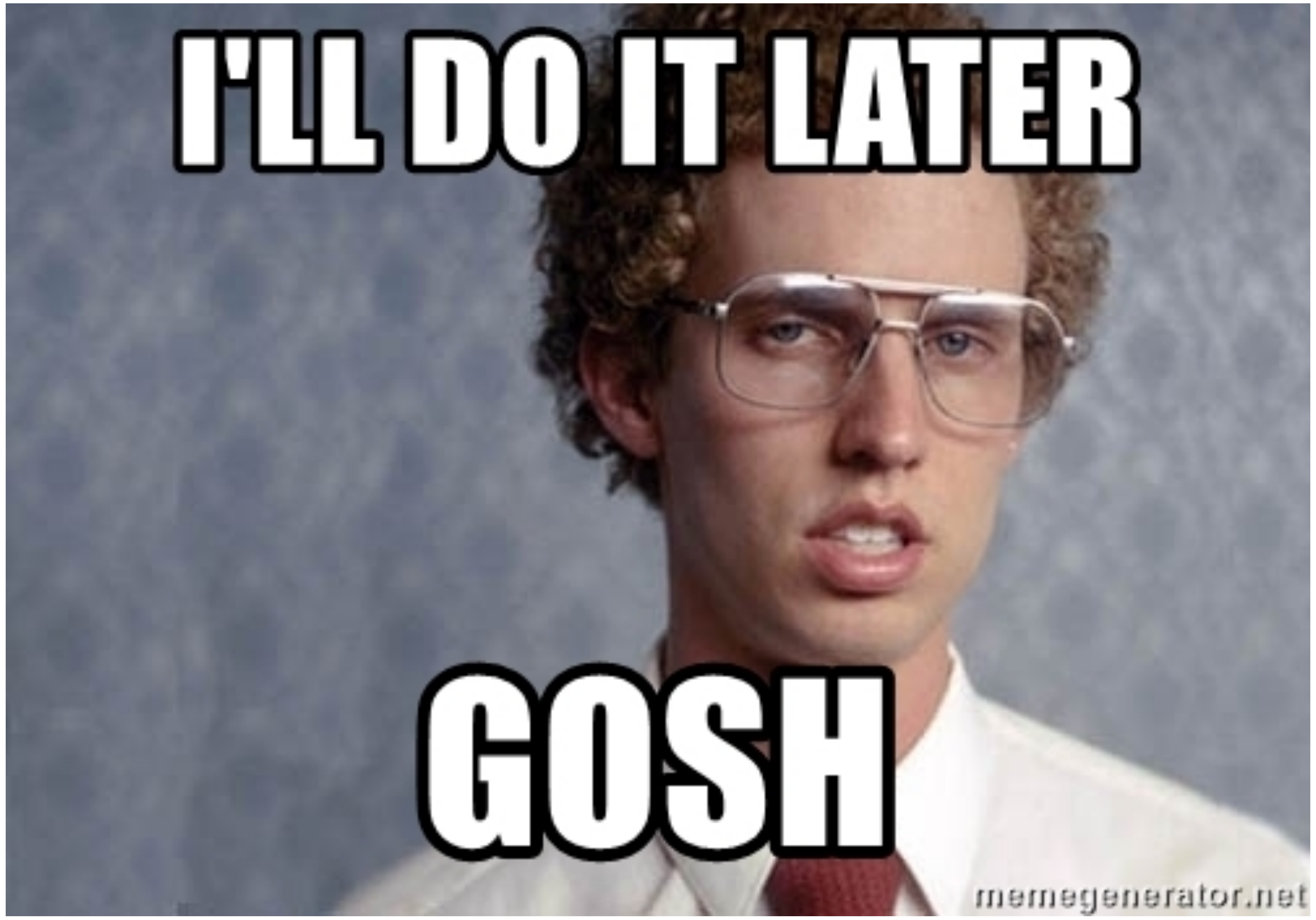
- It takes considerably less effort to deploy an environment
- The environment is defined in code which means software development tools like linting and source control can be applied to infrastructure

Benefits of IaC

- Reduce the risk of change
- Faster Iteration
- More Reliable Systems
- Faster DR
- Improved speed of troubleshooting
- Governance and Security

Self Service Infrastructure

Build First, Automate Later

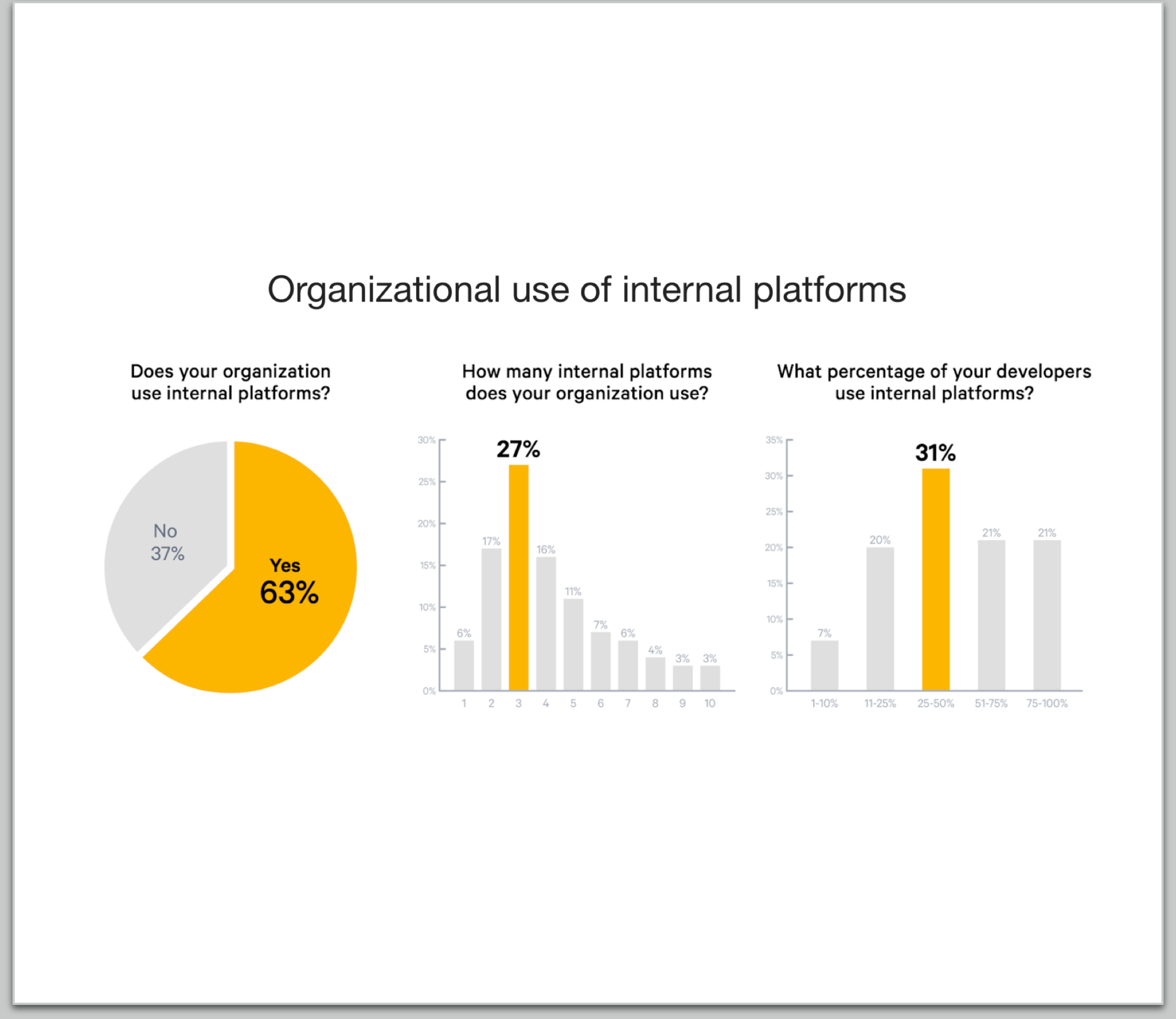


Core Practices of IaC

- Define everything as Code
  - Reusability
  - Consistency
  - Transparency
- Validate and Test
- Small Simple Pieces that you can change independently

State of the DevOps Report 2020

- Internal Platforms Team that develops infrastructure for other teams. Best way to scale DevOps within the organization.
- Responsible for:
  - Infrastructure
  - Environments
  - Deployment Pipelines
  - Self Service Tools for Internal Customers



ARM/CloudFormation or Terraform?

- ARM = IAC for Azure
  AWS = IAC for AWS
- Terraform = IAC for EVERYTHING
- Terraform Concerns?
  - Will I get less functionality with Terraform?
  - Will it cost me an arm and a leg?
  - I only use one cloud right now

Pulumi vs Terraform

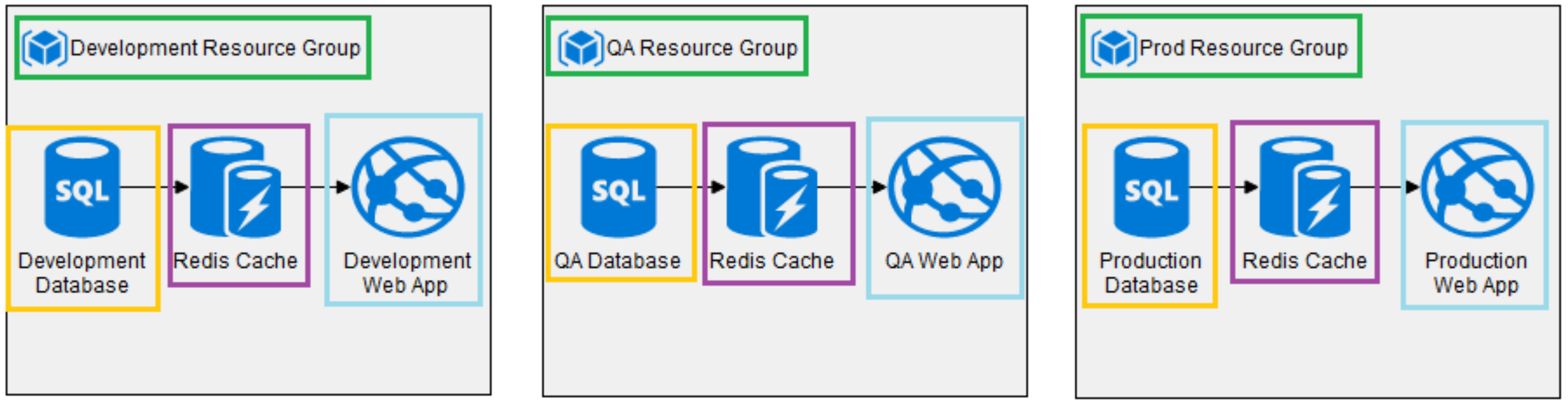
Pulumi

- Use Programming Languages like Python and JavaScript to manage infrastructure
- Harder to transition to from SE role
- Better higher level abstraction

Terraform

- Uses it’s own language called HashiCorp Configuration Language
- Easier to understand and learn for most non-programmers
- Functional limitations, relies on tools like Terragrunt for high level abstraction and functionality

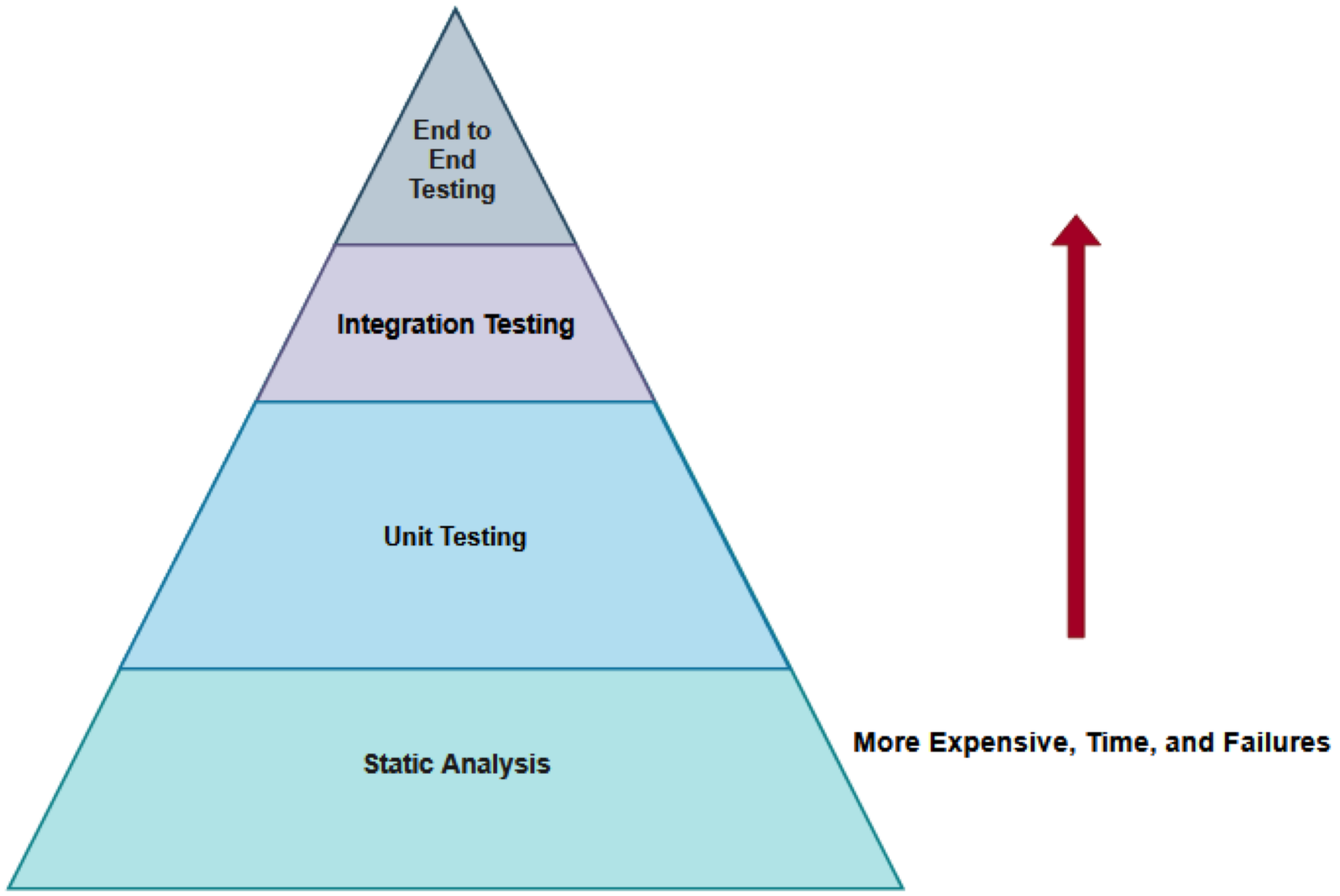
I Don’t Have Time to Write Tests



🟨	SQL Module
🟩	Resource Group Module
🟪	Redis Module
🟦	Web App Module

Types of Tests

- **Static Analysis** - Testing code without running it.
- **Unit Testing** - Testing a single unit.
- **Integration Testing** - Testing the functionality between two or more units.
- **End-to-End Testing** - Testing an entire application infrastructure from the ground up.



Let’s Take a Look at a Terraform Module

Where Do I Start??

- Automate one thing at a time and start slow.
  - IAC for a new environment or application first
  - Automate the most tedious processes first like a server build
- BEWARE: If you try to do everything at once, you will most likely have to go back and re-do it

Q and A?