Infrastructure as Code

Software Architecture

Brae Webb & Richard Thomas

March 16, 2026

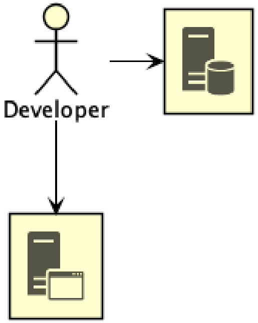
Infrastructure as Code

How did we get here?

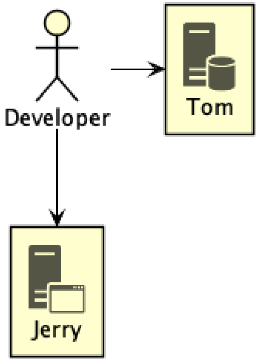
Pre-2000

The *Iron Age*

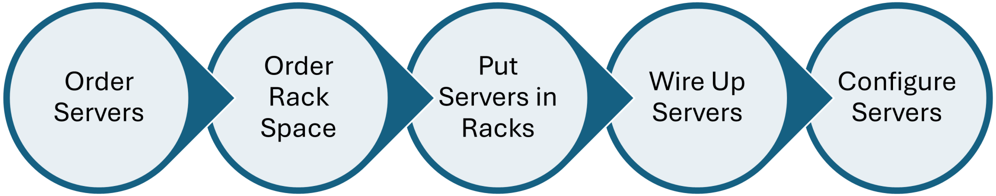
Iron Age



Iron Age



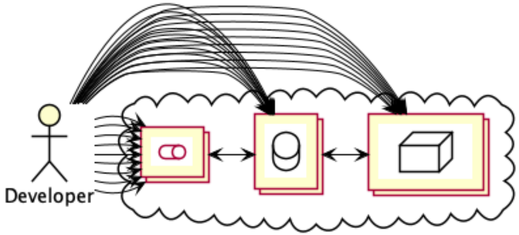
Scaling


Order Servers
Order Rack Space
Put Servers in Racks
Wire Up Servers
Configure Servers

Introducing...

The *Cloud Age*

The Cloud Age



When faced with complexity

Automate it!

The larger story

Server Config Config Management

The larger story

Server Config	Config Management
Application Config	Config Files

The larger story

Server Config	Config Management
Application Config	Config Files
Provisioning	Infrastructure Code

The larger story

Server Config	Config Management
Application Config	Config Files
Provisioning	Infrastructure Code
Building	Continuous Integration

The larger story

Server Config	Config Management
Application Config	Config Files
Provisioning	Infrastructure Code
Building	Continuous Integration
Deployment	Continuous Deployment

The larger story

Server Config	Config Management
Application Config	Config Files
Provisioning	Infrastructure Code
Building	Continuous Integration
Deployment	Continuous Deployment
Testing	Automated Tests

The larger story

Server Config	Config Management
Application Config	Config Files
Provisioning	Infrastructure Code
Building	Continuous Integration
Deployment	Continuous Deployment
Testing	Automated Tests
Database Administration	Schema Migration

The larger story

Server Config	Config Management
Application Config	Config Files
Provisioning	Infrastructure Code
Building	Continuous Integration
Deployment	Continuous Deployment
Testing	Automated Tests
Database Administration	Schema Migration
Specifications	Behaviour Driven Development

Definition 0. Infrastructure Code

Code that provisions and manages *infrastructure resources*.

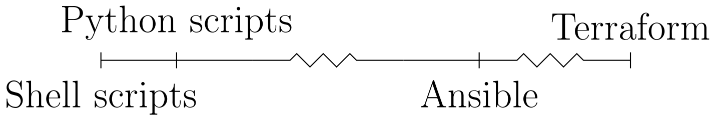
Definition 0. Infrastructure Code

Code that provisions and manages *infrastructure resources*.

Definition 0. Infrastructure Resources

Compute resources, networking resources, and storage resources.

Infrastructure Code



Shell Scripts

```bash
#!/bin/bash

SG=$(aws ec2 create-security-group ...)

aws ec2 authorize-security-group-ingress --group-id "$SG"

INST=$(aws ec2 run-instances --security-group-ids "$SG" \
        --instance-type t2.micro)
```

Python

```python
import boto3

def create_instance():
    ec2_client = boto3.client("ec2", region_name="us-east-1")
    response = ec2.create_security_group(...)
    security_group_id = response['GroupId']

    data = ec2.authorize_security_group_ingress(...)

    instance = ec2_client.run_instances(
        SecurityGroups=[security_group_id],
        InstanceType="t2.micro",
        ...
    )
```

Terraform

```
resource "aws_instance" "hextris-server" {
    instance_type = "t2.micro"
    security_groups = [aws_security_group.hextris-server.name]
    ...
}

resource "aws_security_group" "hextris-server" {
    ingress {
        from_port = 80
        to_port = 80
        ...
    }
    ...
}
```

Question

Notice anything different?

The main difference

Imperative vs. Declarative

Declarative IaC

- Define your *desired* infrastructure state
  - as code
- Engine interprets difference between the *desired* and *actual* state
  - Modifying infrastructure to deliver *desired* state

Infrastructure Code

- Provisions and manages *infrastructure resources*.

Infrastructure Code

- Provisions and manages *infrastructure resources*.
- Only one part of the movement to *automate* the complexities of development.

Infrastructure Code

- Provisions and manages *infrastructure resources*.
- Only one part of the movement to *automate* the complexities of development.
- Ranges from simple shell scripts up to. . . ?

Infrastructure Code

- Provisions and manages *infrastructure resources*.
- Only one part of the movement to *automate* the complexities of development.
- Ranges from simple shell scripts up to…?
- Tendency to be *declarative*.

Typo?

Infrastructure Code $\neq$ Infrastructure *as* Code

Definition 0. Infrastructure as Code

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

Warning!

Infrastructure as Code still *early* and quite *bad*.

Question

What are *good coding practices*?

Good Coding Practice #1

Everything as Code

```bash
#!/bin/bash

./download-dependencies
./build-resources
cp -r output/* artifacts/
```

```bash
#!/bin/bash

./download-dependencies
./build-resources
cp -r output/* artifacts/
```

```
$ cp: directory artifacts does not exist
```

```
resource "aws_instance" "hextris-server" {
    instance_type = "t2.micro"
    security_groups = ["sg-6400"]
    ...
}
```

```
resource "aws_instance" "hextris-server" {
    instance_type = "t2.micro"
    security_groups = [aws_security_group.hextris-server.name]
    ...
}

resource "aws_security_group" "hextris-server" {
    ingress {
        from_port = 80
        to_port = 80
        ...
    }
    ...
}
```

Everything as code avoids

Configuration drift

Configuration drift creates

Snowflakes

Benefits

1. Reproducible

Good Coding Practice #2

Version Control

Benefits

1. Restorable
2. Accountable

Good Coding Practice #3

Automation

Benefits

1. Consistent

Good Coding Practice #4

Code Reuse

Benefits

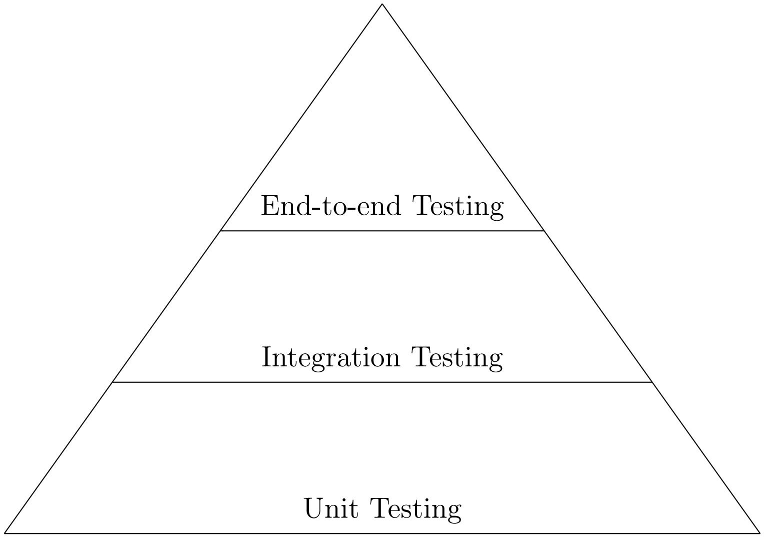
1. Better[1] code
2. Less work
3. Only one place to update (or verify)

[1] generally

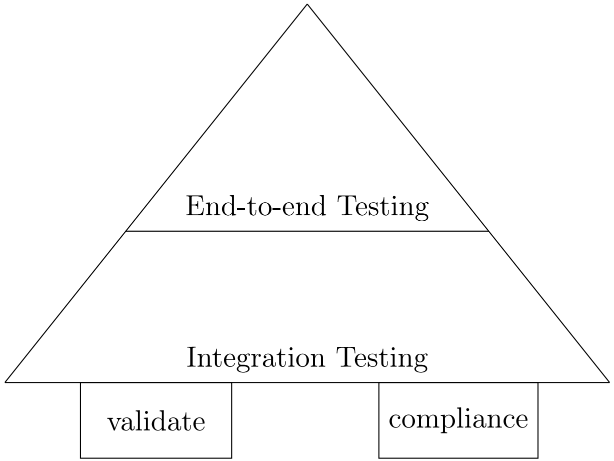
Good Coding Practice #5

Testing

Test Pyramid



IaC Test Pyramid



```go
func TestTerraformAwsInstance(t *testing.T) {
    terraformOptions := terraform.WithDefault(t, &terraform.Options{
        TerraformDir: "../week03/",
    })

    defer terraform.Destroy(t, terraformOptions)
    terraform.InitAndApply(t, terraformOptions)

    publicIp := terraform.Output(t, terraformOptions, "public_ip")
    url := fmt.Sprintf("http://%s:8080", publicIp)

    http_helper.HttpGetWithCustomValidation(t, url, nil, 200,
        func(code, resp) { code == 200 &&
                              strings.Contains(resp, "hextris")})
}
```

```gherkin
Feature: Define AWS Security Groups

Scenario: Only selected ports should be publicly open
    Given I have AWS Security Group defined
    When it contains ingress
    Then it must only have tcp protocol and port 22,443 for 0.0.0.0/0
```

Benefits

1. Trust

Prac Next Week

Learn how to use Terraform to write IaC and deploy resources on AWS.