Infrastructure as Code

 $Software\ Architecture$

Brae Webb

March 20, 2023

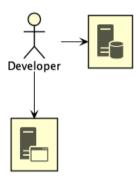
Infrastructure as Code

How did we get here?

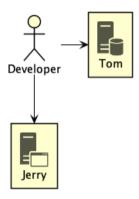
$Pre \hbox{-} 2000$

The *Iron Age*

$Iron\ Age$



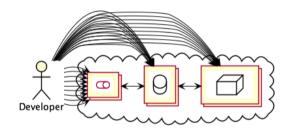
$Iron\ Age$



Introducing...

The Cloud Age

The Cloud Age



When faced with complexity

Automate it!

Server Config Config Management

Server Config Config Management Application Config Config Files

Server Config Config Management
Application Config Config Files
Provisioning Infrastructure Code

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

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

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

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

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 1. Infrastructure Code

Code that provisions and manages *infrastructure resources*.

Definition 2. Infrastructure Code

Code that provisions and manages $infrastructure\ resources$.

Definition 3. Infrastructure Resources

Compute resources, networking resources, and storage resources.

Infrastructure Code

Python scripts Terraform
Shell scripts Ansible

```
#!/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)

```
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(
10
           SecurityGroups=[security_group_id],
11
           InstanceType="t2.micro".
12
13
           . . .
14
```

```
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
10
11
           . . .
12
13
        . . .
14
```

Question

Notice anything different?

The main difference

Imperative vs. declarative

$Infrastructure\ Code$

• 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...?
- Tendancy to be *declarative*.

Typo?

Infrastructure Code \neq Infrastructure as Code

Definition 4. 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

```
#!/bin/bash

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

```
#!/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
10
11
           . . .
12
13
        . . .
14
```

Everything as code avoids

Configuration drift

$Configuration\ drift\ creates$

Snowflakes

1. Reproducible.

Good Coding Practice #2 Version control

- 1. Restorable.
- 2. Accountable.

Good Coding Practice #3 Automation

1. Consistent.

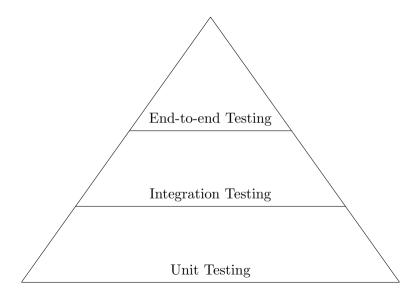
Good Coding Practice #4 Code Reuse

- 1. Better¹ code.
- 2. Less work.
- 3. Only one place to update (or verify).

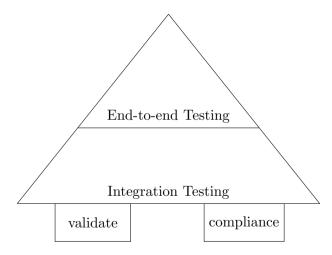
¹generally

Good Coding Practice #5 Testing

Test Pyramid



IaC Test Pyramid



```
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)
10
       http_helper.HttpGetWithCustomValidation(t, url, nil, 200,
12
           func(code, resp) { code == 200 &&
13
                              strings.Contains(resp, "hextris")})
14
15
```

Feature: Define AWS Security Groups

Scenario: Only selected ports should be publicly open

When it contains ingress

Given I have AWS Security Group defined

Then it must only have tcp protocol and port 22,443 for 0.0.0.0/0

3

1. Trust.