CSSE6400

### **Brae Webb**

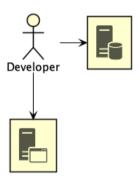
March 21, 2022

# How did we get here?

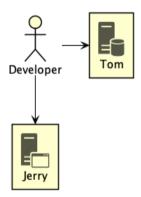
Pre-2000

# The Iron Age

#### Iron Age



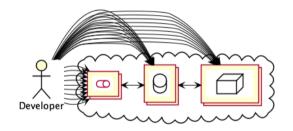
#### Iron Age



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

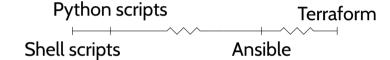
Code that provisions and manages infrastructure resources.

Definition 2. Infrastructure Code

Code that provisions and manages infrastructure resources.

Definition 3. Infrastructure Resources

Compute resources, networking resources, and storage resources.



```
#!/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)

8

```
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(...)
8
       instance = ec2 client.run instances(
1.0
          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. Declaritive

• Provisions and manages *infrastructure resources*.

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

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

- 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 declaritive.

Typo?

Infrastructure Code  $\neq$  Infrastructure  $\alpha s$  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.

### Version control

- 1. Restorable.
- 2. Accountable.

### **Automation**

### 1. Consistent.

## Code Reuse

- 1. Better<sup>1</sup> code.
- 2. Less work.

# **Testing**

# 1. Trust.