#DEVOPSDAYS

Bringing software development practices to your infrastructure

@jennapederson

Fun Fact

I once had the phrase "automated test fanatic" on my business card.

The awesomeness of Infrastructure as Code

@jennapederson

(or account or region)

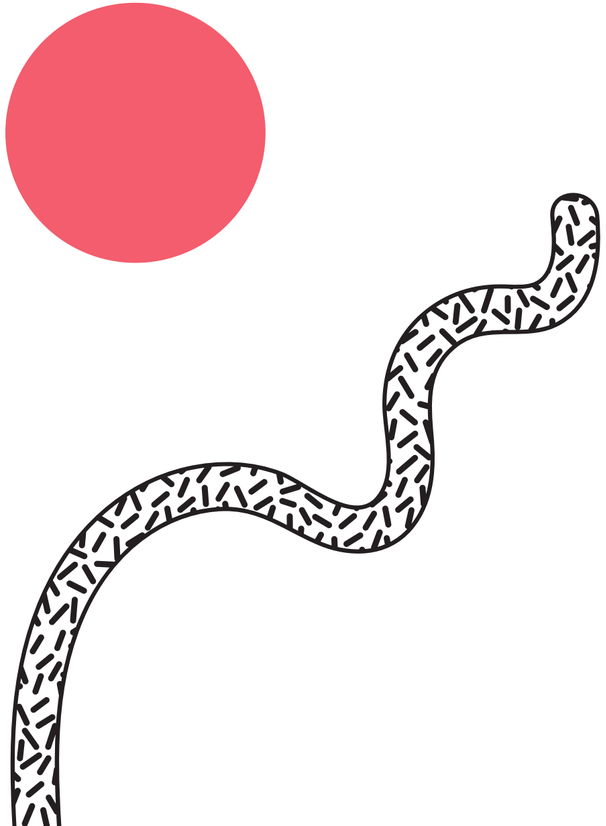
@jennapederson

Infrastructure as Code IS Code

```
3     super(scope, id, props);
4
5     const taskDefinition = new ecs.FargateTaskDefinition(this, 'TaskDe
6       memoryLimitMiB: 1024,
7       cpu: 256,
8     });
9
10    const image = props.image || new ecs.AssetImage(path.join(__dirnam
11    const container = taskDefinition.addContainer("WebServer", {
12      image,
13    });
14    container.addPortMappings({containerPort: 80});
15
16    const service = new ecs.FargateService(this, 'Service', {
17      cluster: props.cluster,
18      taskDefinition,
19    });
20
21    const lb = new elbv2.ApplicationLoadBalancer(this, 'LB', {
22      vpc: props.vpc,
23      internetFacing: true,
24    });
25
26    const listener = lb.addListener('HttpListener', {
27      port: 80,
28    });
```

@jennapederson

Agenda



Different Types of Testing

Using the right type at the right time

Using Test Driven Development

Build what you need and only what you need

Testing Your Infrastructure Directly

Making sure it was created correctly and hasn't drifted

Using a CI/CD Pipeline

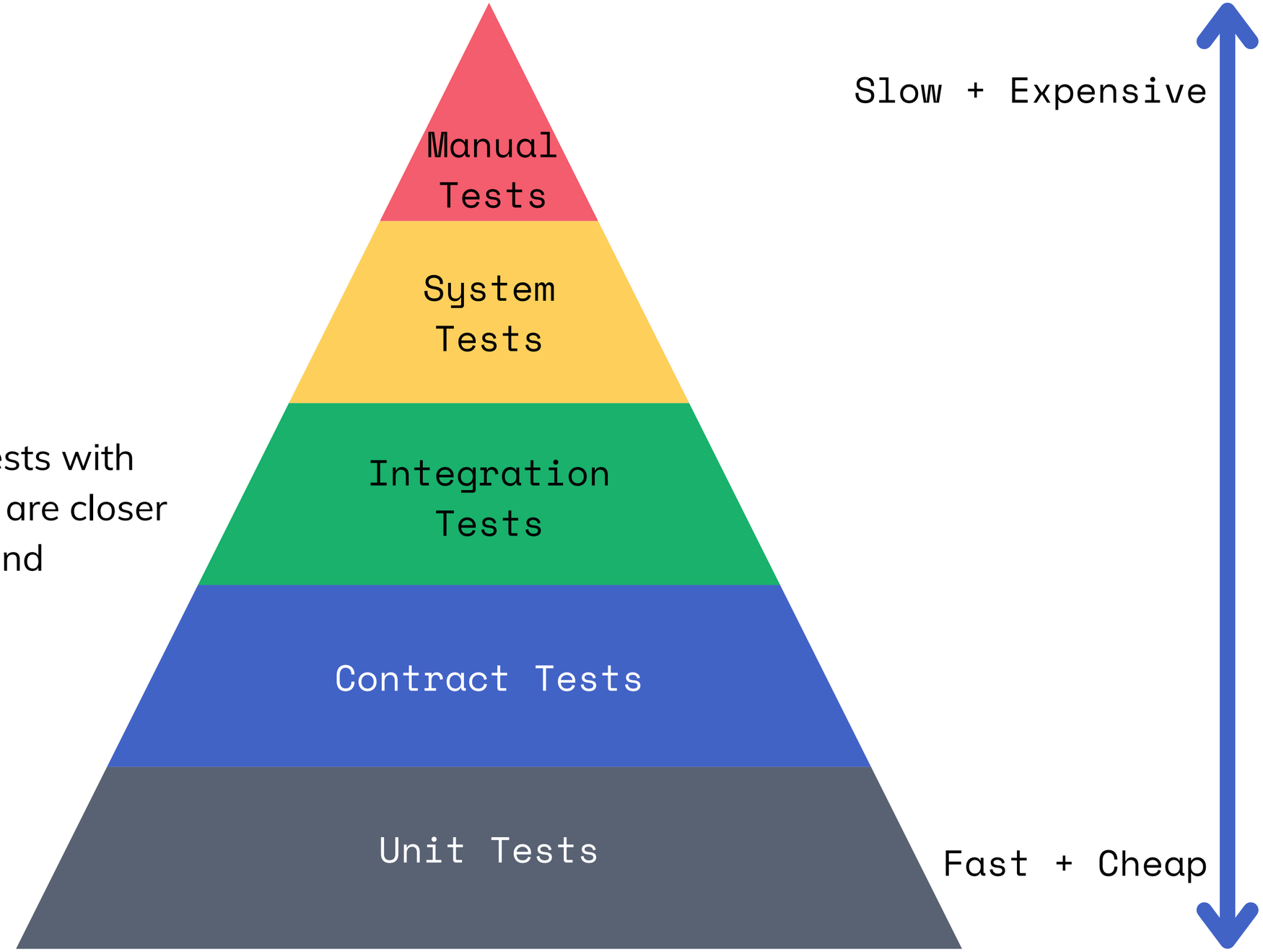
Run tests in the real world and isolate issues quicker

Why Test Infrastructure?

The cloud makes it easier and quicker to provision infrastructure, but there is complexity with that scale.

Failing Fast

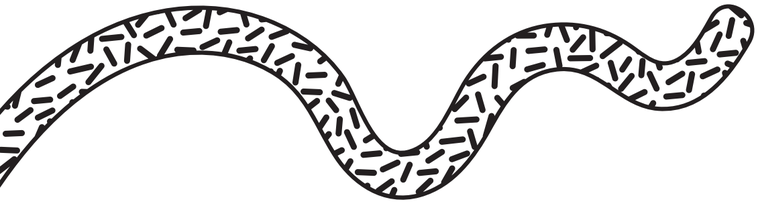
Balance fast and cheap tests with more expensive tests that are closer to the real infrastructure and production environment.



If you're TDDing your application code, why not do the same for your infrastructure code?

Benefits of TDD

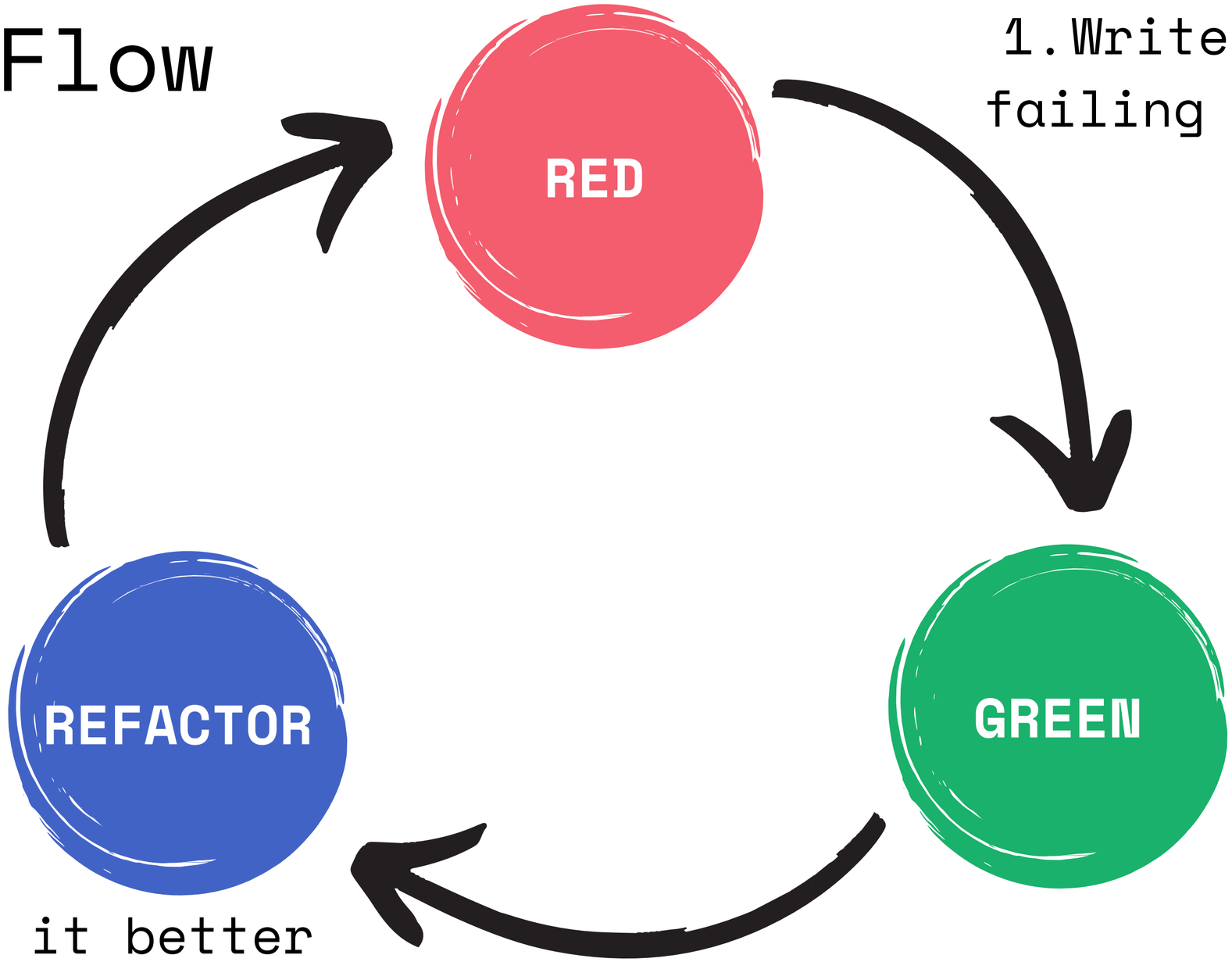
- Reduced defect rates
- Improve the overall design
- Focused on requirements
- Focused on small chunks
- Serves as documentation



Confidence!

@jennapederson

The Flow



1. Write a failing test

2. Write only enough code to make it pass

3. Make it better

What is a Unit Test?

Exercises a small part of your application, one unit, and verifies that it's correct.

Isolated from other resources and external APIs, reducing the scope and the number of variables that can affect the results.

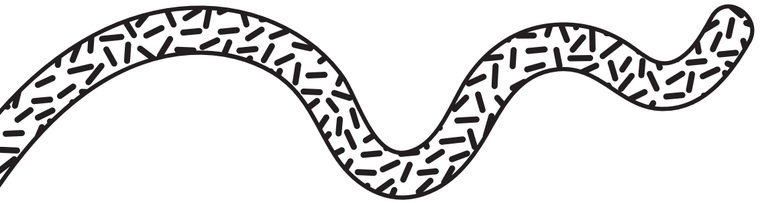
Unit Testing Infrastructure Code

Code. Not infrastructure.

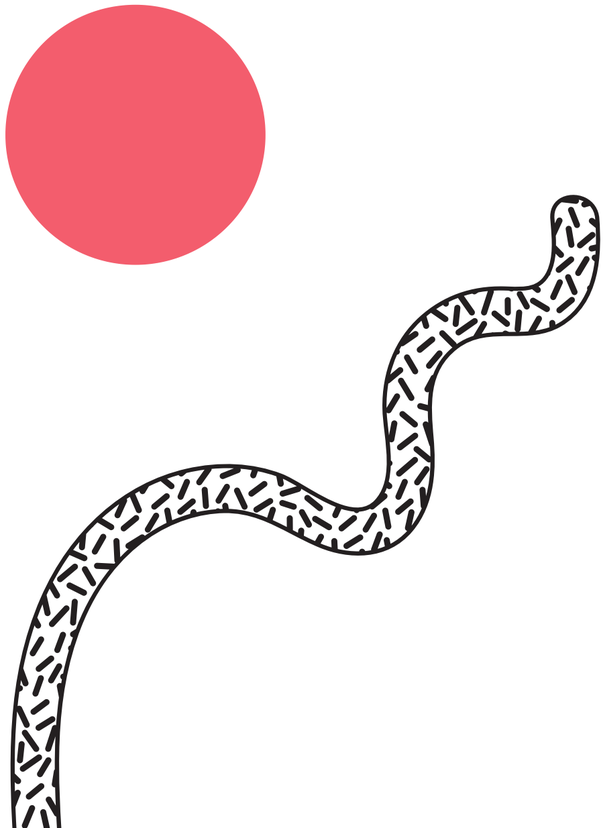
@jennapederson

A unit test checks:

- If a resource will be created with the correct configuration
- The correct number of resources will be created
- Dependencies between resources are correct
- Interpolated values are correct



Why unit tests?



- Cheap to write, cheap to run
- Get feedback early on to shorten the feedback loop between changes
- Serves as documentation
- Can be run in your CI/CD tool

Demo

S3 + CDK + Jest

infrastructure-test-examples

EXPLORER

INFRASTRUCTUR...

- bin
- cdk.out
- inspec
- lib
- node_modules
- test
- .gitignore
- .npmignore
- cdk.json
- jest.config.js
- LICENSE
- package-lock.json
- package.json
- README.md
- tsconfig.json

OUTLINE

TIMELINE

NPM SCRIPTS

Show All Commands	⇧ ⌘ P
Go to File	⌘ P
Find in Files	⇧ ⌘ F
Start Debugging	F5
Toggle Terminal	^ `

⊗ 2 ⚠ 0 ✓ jsonc | ✓ tsconfig.json

How do we go from code to infrastructure?

```
3     super(scope, id, props);
4
5     const taskDefinition = new ecs.FargateTaskDefinition(this, 'TaskDe
6       memoryLimitMiB: 1024,
7       cpu: 256,
8     });
9
10    const image = props.image || new ecs.AssetImage(path.join(__dirnam
11    const container = taskDefinition.addContainer("WebServer", {
12      image,
13    });
14    container.addPortMappings({containerPort: 80});
15
16    const service = new ecs.FargateService(this, 'Service', {
17      cluster: props.cluster,
18      taskDefinition,
19    });
20
21    const lb = new elbv2.ApplicationLoadBalancer(this, 'LB', {
22      vpc: props.vpc,
23      internetFacing: true,
24    });
25
26    const listener = lb.addListener('HttpListener', {
27      port: 80
```

InfrastructureTestExamplesStack

Delete | Update | Sta

Stack info | **Events** | Resources | Outputs | Parameters | Template | Change sets

Events (100+)

Search events

Timestamp	Logical ID	Status	Status reason
2021-07-17 20:41:34 UTC-0500	InfrastructureTestExamplesStack	UPDATE_COMPLETE	-
2021-07-17 20:41:33 UTC-0500	InfrastructureTestExamplesStack	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-
2021-07-17 20:41:05 UTC-0500	InfrastructureTestExamplesStack	UPDATE_IN_PROGRESS	User Initiated
2021-07-17 20:01:16 UTC-0500	InfrastructureTestExamplesStack	UPDATE_COMPLETE	-
2021-07-17 20:01:16 UTC-0500	InfrastructureTestExamplesStack	UPDATE_COMPLETE_CLEANUP_IN_PROGRES	-

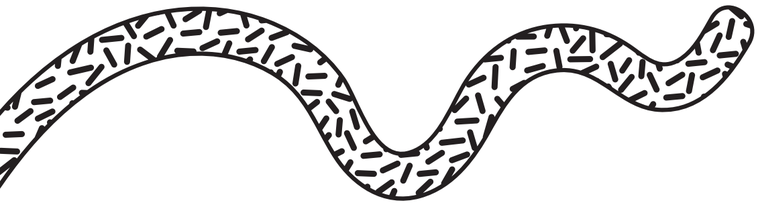
What is an Integration Test?

Tests the interactions across different units or modules, or in the case of infrastructure testing, across cloud resources.

Verifies your provisioned cloud resources are created and configured as you expect them to be.

Gives you confidence in infrastructure at scale and at velocity.

Chef InSpec



- Open-source framework to test and audit cloud resources IN the cloud
- Tests are written with a DSL
- Can be used across teams
- Test resources that are managed manually or with code
- Ensures requirements are met at every stage of the SDLC

Demo

EC2 + RDS + CDK + InSpec

infrastructure-test-examples

EXPLORER

INFRASTRUCTURE-TEST-EXAMPLES

- bin
- cdk.out
- inspec
- lib
- node_modules
- test
- .gitignore
- .npmignore
- cdk.json
- jest.config.js
- LICENSE
- package-lock.json
- package.json
- README.md
- tsconfig.json

OUTLINE

NPM SCRIPTS

Show All Commands	⇧ ⌘ P
Go to File	⌘ P
Find in Files	⇧ ⌘ F
Start Debugging	F5
Toggle Terminal	^ `

⊗ 0 ⚠ 0 ✓ plaintext | ✓ 01_ec2_instance.rb.example

Detecting Drift

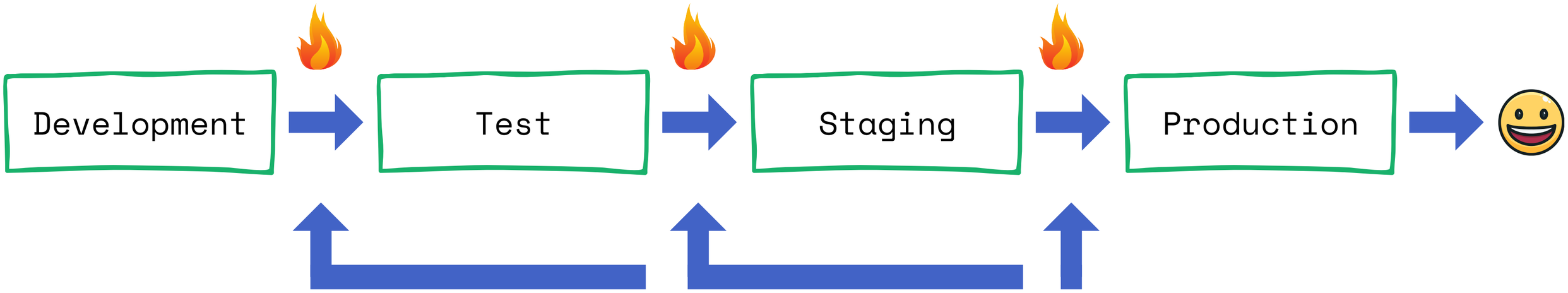
Use InSpec to compare the desired state
with the actual state of your
cloud resources.

Can be used against any resources,
regardless of how they are managed.

Without CI/CD

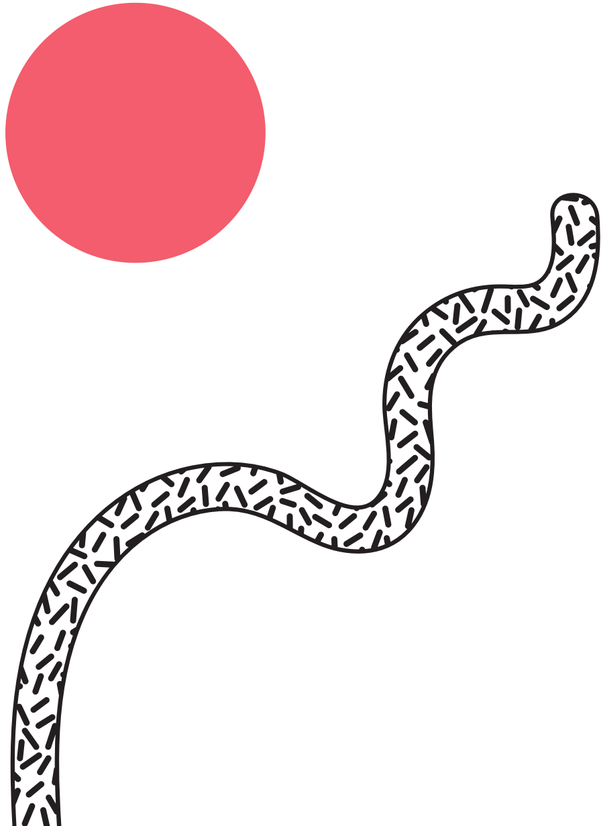
@jennapederson

With CI/CD



@jennapederson

Wrapping Up



Infrastructure code is like any other code, treat it as such.

Testing is never done, even once you reach production.

It’s cheaper to detect broken code early.

@jennapederson

Thank you!

@jennapederson

/in/jennapederson

jennapederson

https://jenna.link/hq7

Feedback