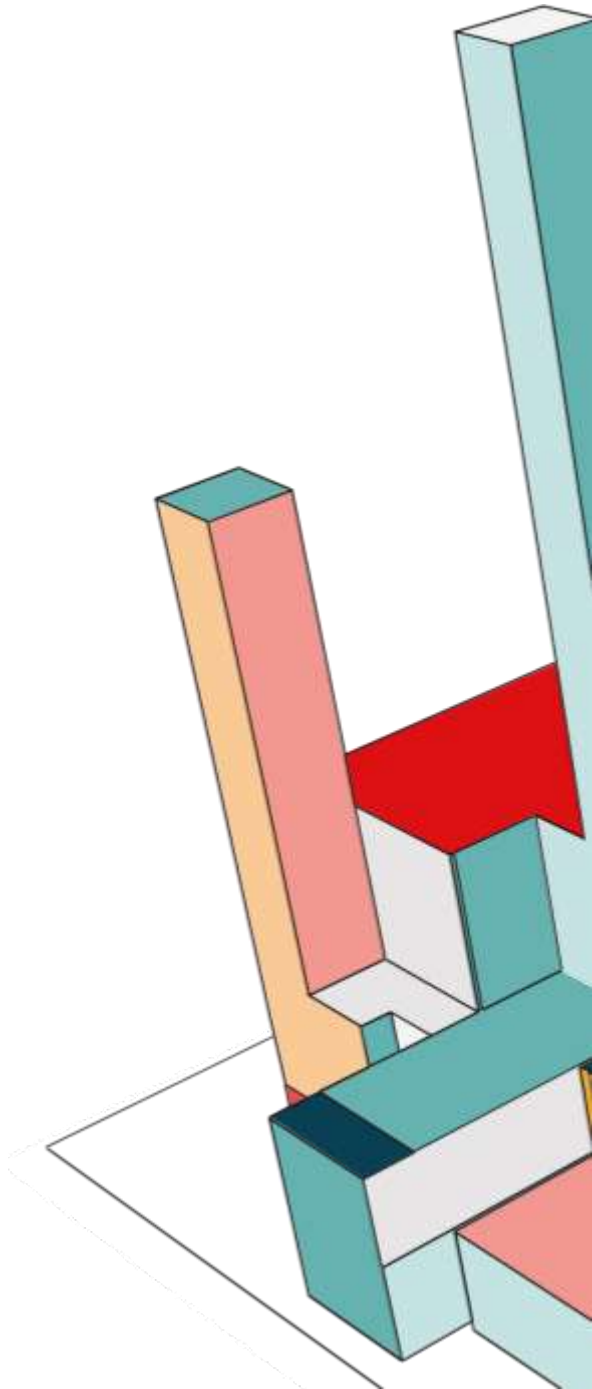
BUILDING CI/CD PIPELINES FOR INFRASTRUCTURE

FROM AUTOMATION TO TESTING

BY RAJESHWARI VAKHARIA

AGENDA

- Introduction
- Introducing IaC to CI/CD Pipeline
- Organizing the infra repo
- Testing the Stack



CHALLENGES WITH DECLARATIVE LANGUAGES

- TESTS FOR DECLARATIVE CODE OFTEN HAVE LOW VALUE

```
subnet:
  name: private_A
  address_range: 192.168.0.0/16
```

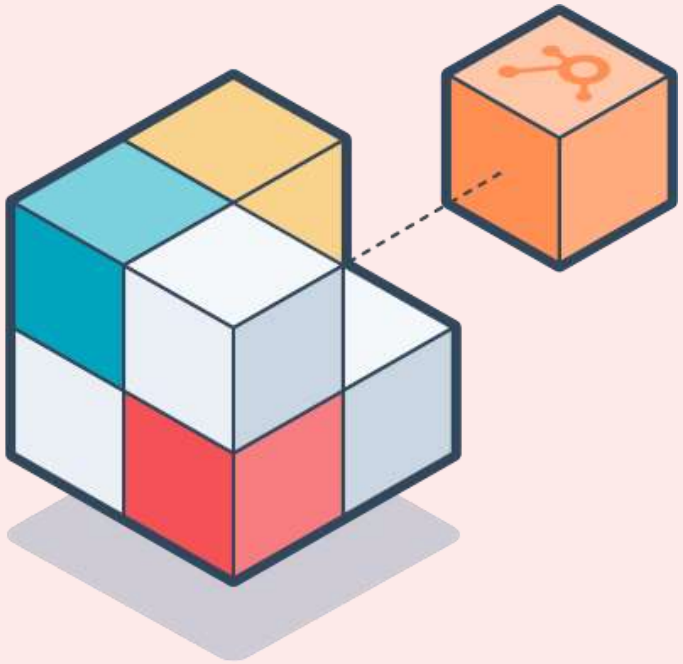
A test for this would simply restate the code:

```
assert:
  subnet("private_A").exists
assert:
  subnet("private_A").address_range is("192.168.0.0/16")
```

- A SUITE OF LOW-LEVEL TESTS OF DECLARATIVE CODE CAN BECOME A BOOKKEEPING EXERCISE.
- THE TEST RATHER BECOMES A CHECK.

- TESTING INFRASTRUCTURE CODE IS SLOW
- DEPENDENCIES COMPLICATE INFRASTRUCTURE

WAYS TO OVERCOME THESE CHALLENGES



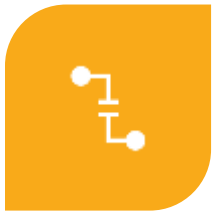
- Divide infrastructure into tractable pieces
- Clarify, minimize, and isolate dependencies
- Progressive Testing
- Choice of Ephemeral or Persistent Instances
- Online and Offline Tests

INTRODUCING IAC TO CI/CD PIPELINE

- The delivery pipeline combines progressive testing with the delivery of the infrastructure to different environments to the path of production.

AUTOMATE INFRASTRUCTURE PROVISIONING END-TO-END

ELIMINATE MANUAL COMMAND EXECUTION

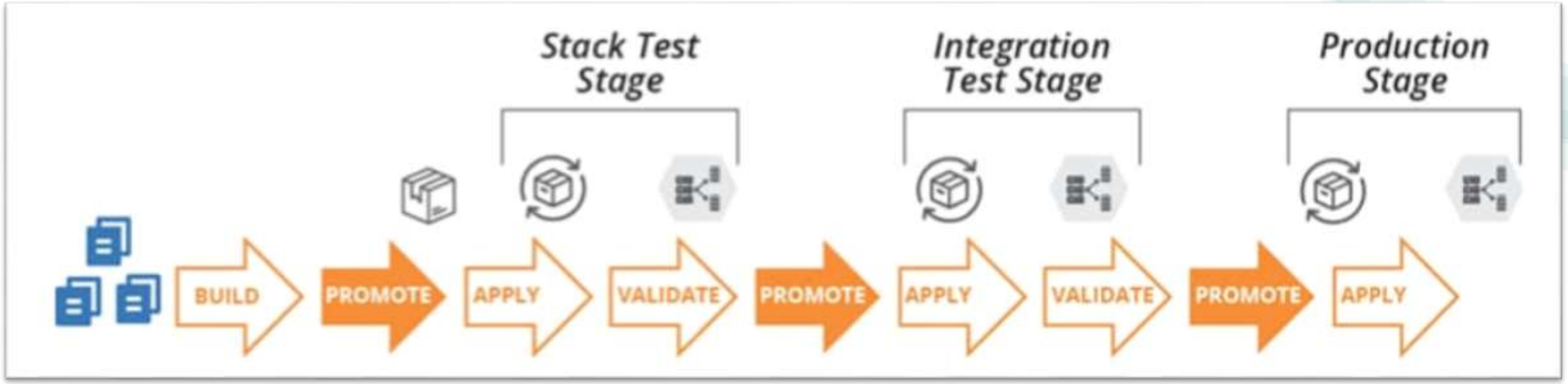


MAINTAIN VERSION-CONTROLLED INFRASTRUCTURE THE CODE

ENABLE ON-DEMAND AND REPEATABLE DEPLOYMENTS

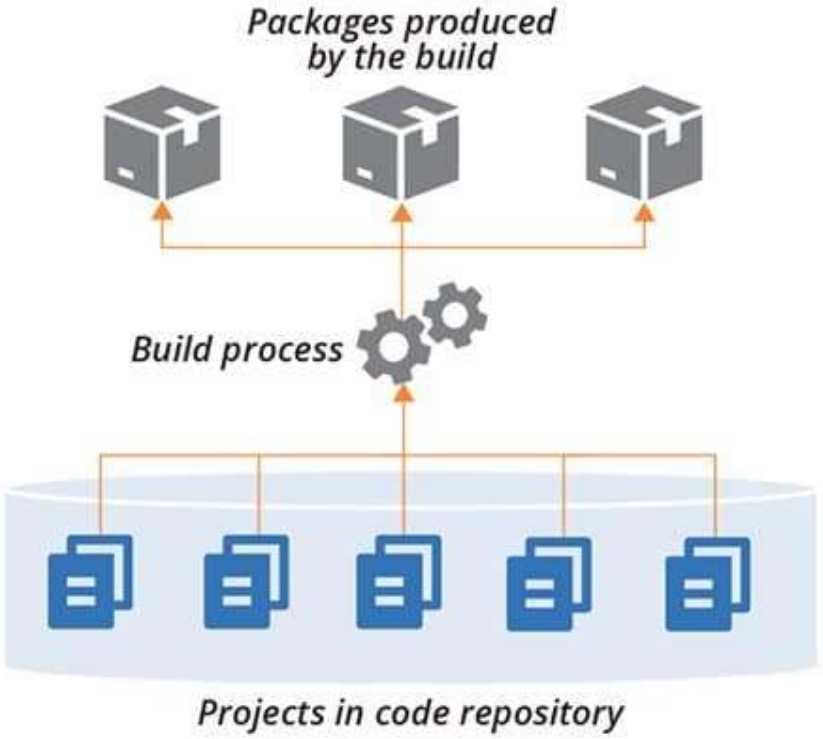
INTEGRATE AUTOMATED TESTING FOR INFRASTRUCTURE CHANGES

ENHANCE COLLABORATION BETWEEN TEAMS

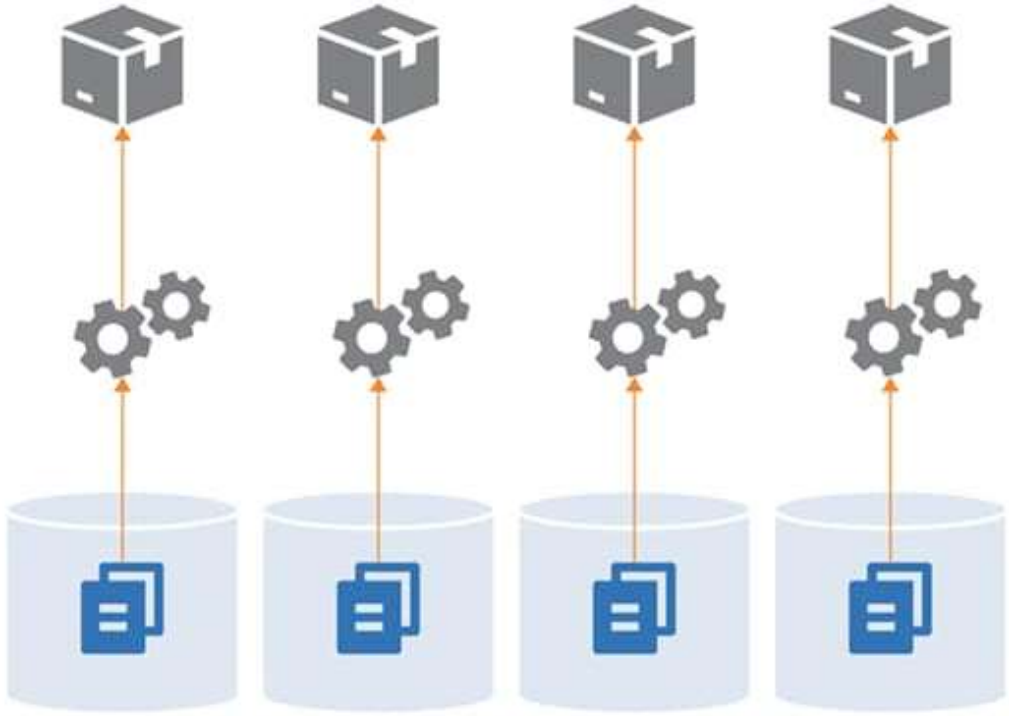


INFRASTRUCTURE CODE PROJECT DELIVERY PHASES

ORGANIZING THE REPOSITORY



Mono Repo- One Repo, Multiple Builds



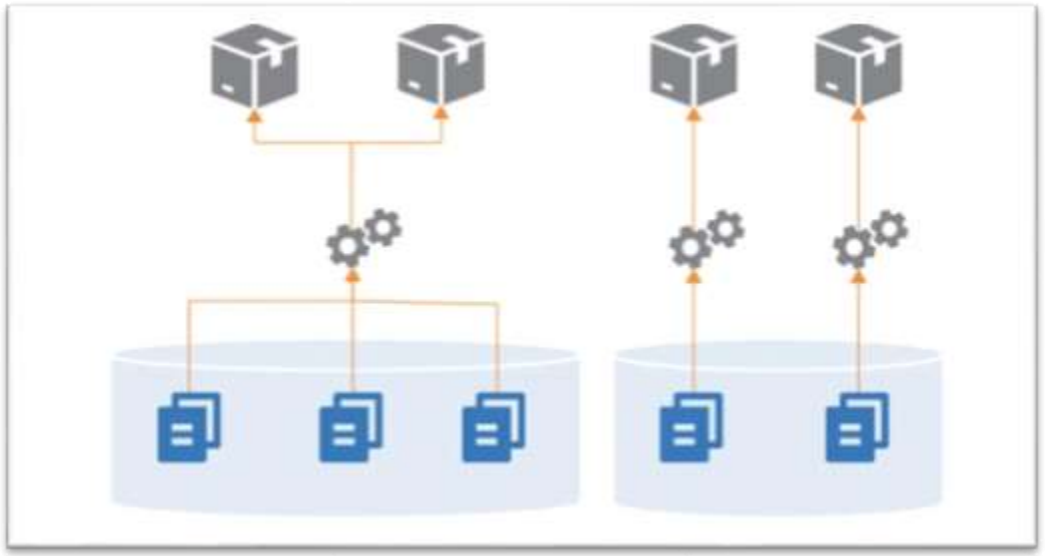
Micro Repo- Multiple Repo Multiple Builds

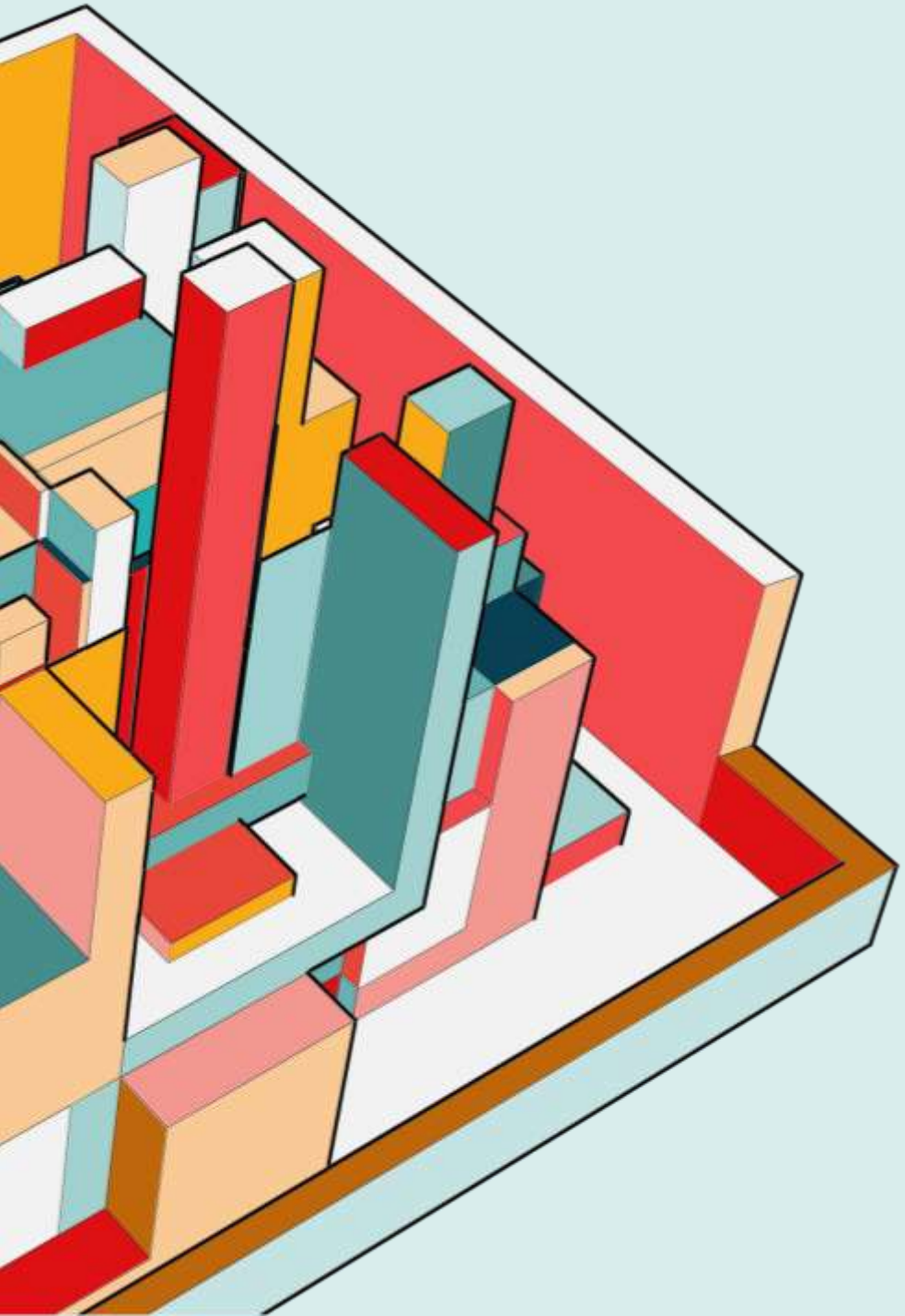
- In practice, it’s more practical to build across multiple projects in a single repository, because their code is versioned together.

- Pushing changes for a single build to multiple repositories complicates the delivery process.

- Splitting repositories can be needed when:
  - It grows in size
  - History
  - Number of users
  - Activity level

MULTIPLE REPO, MULTIPLE PROJECTS

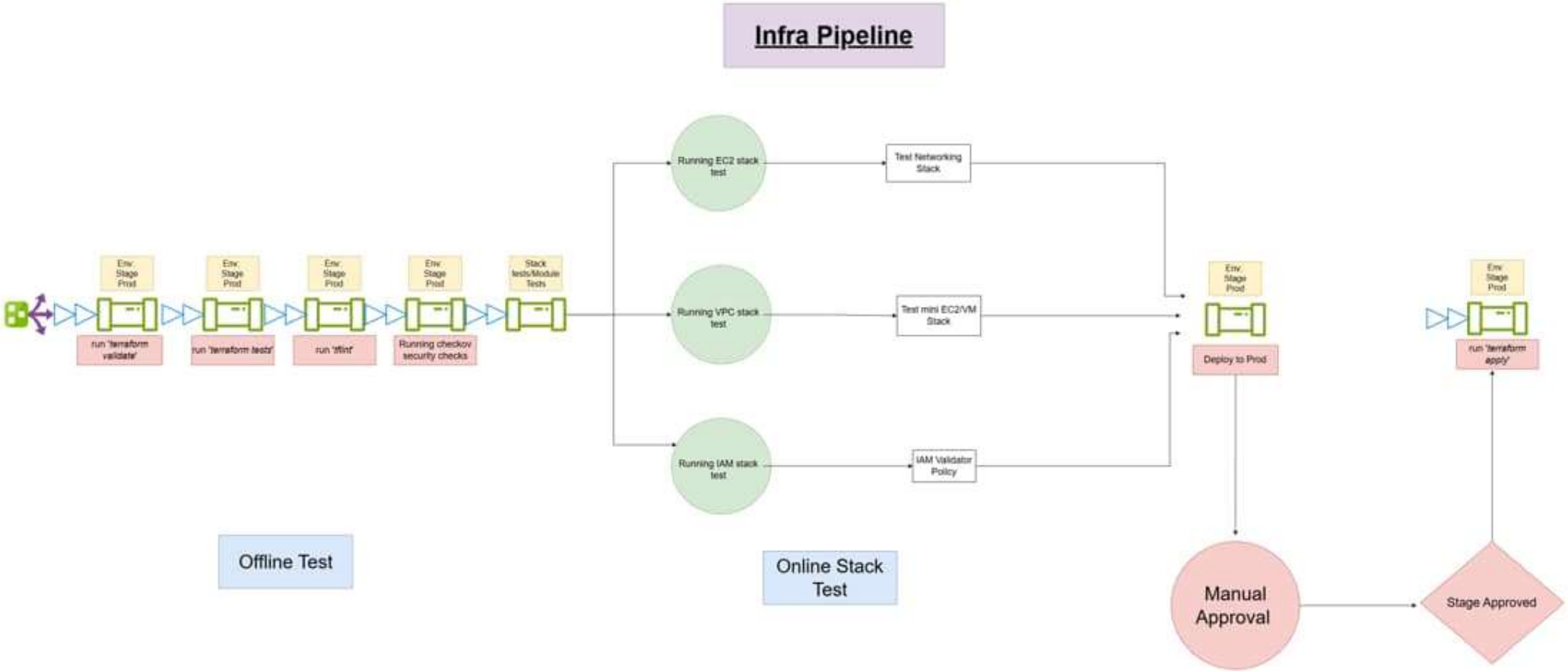




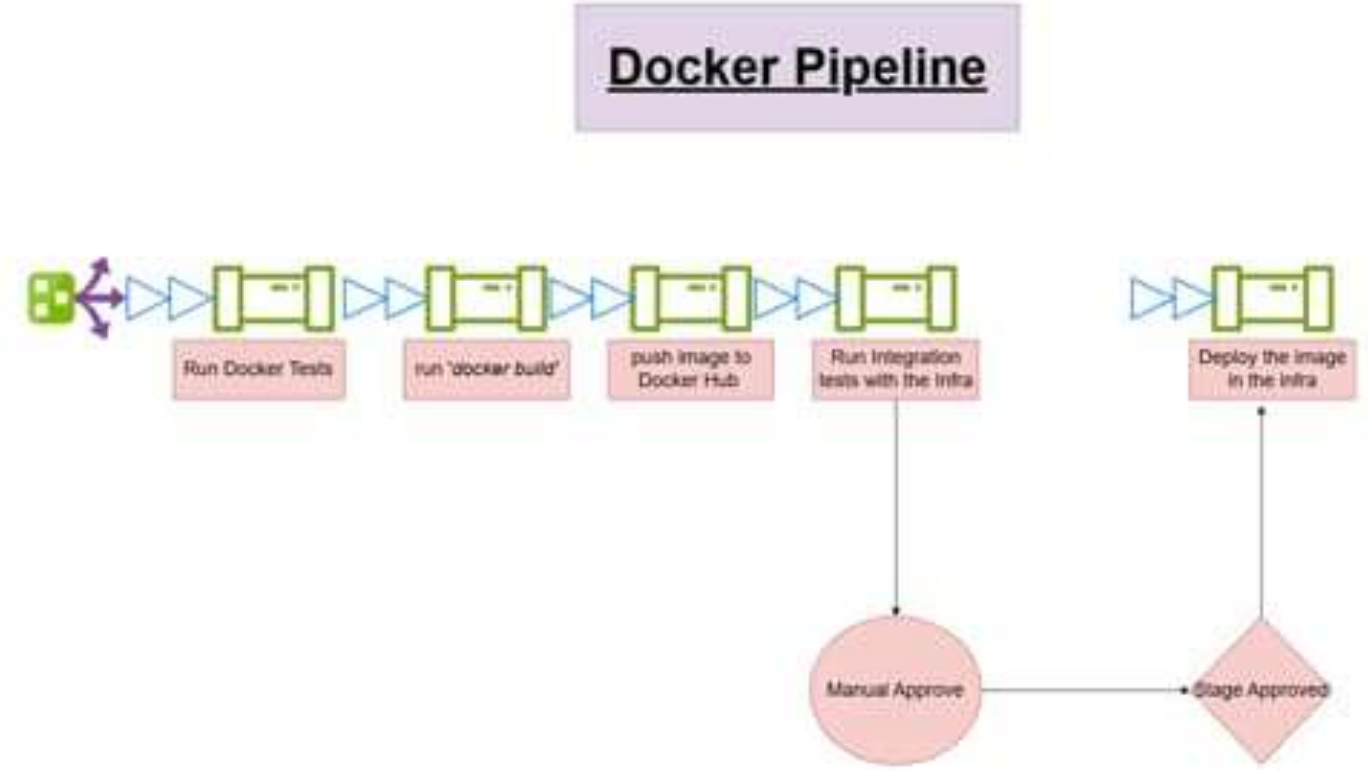
PRACTICAL IMPLEMENTATION

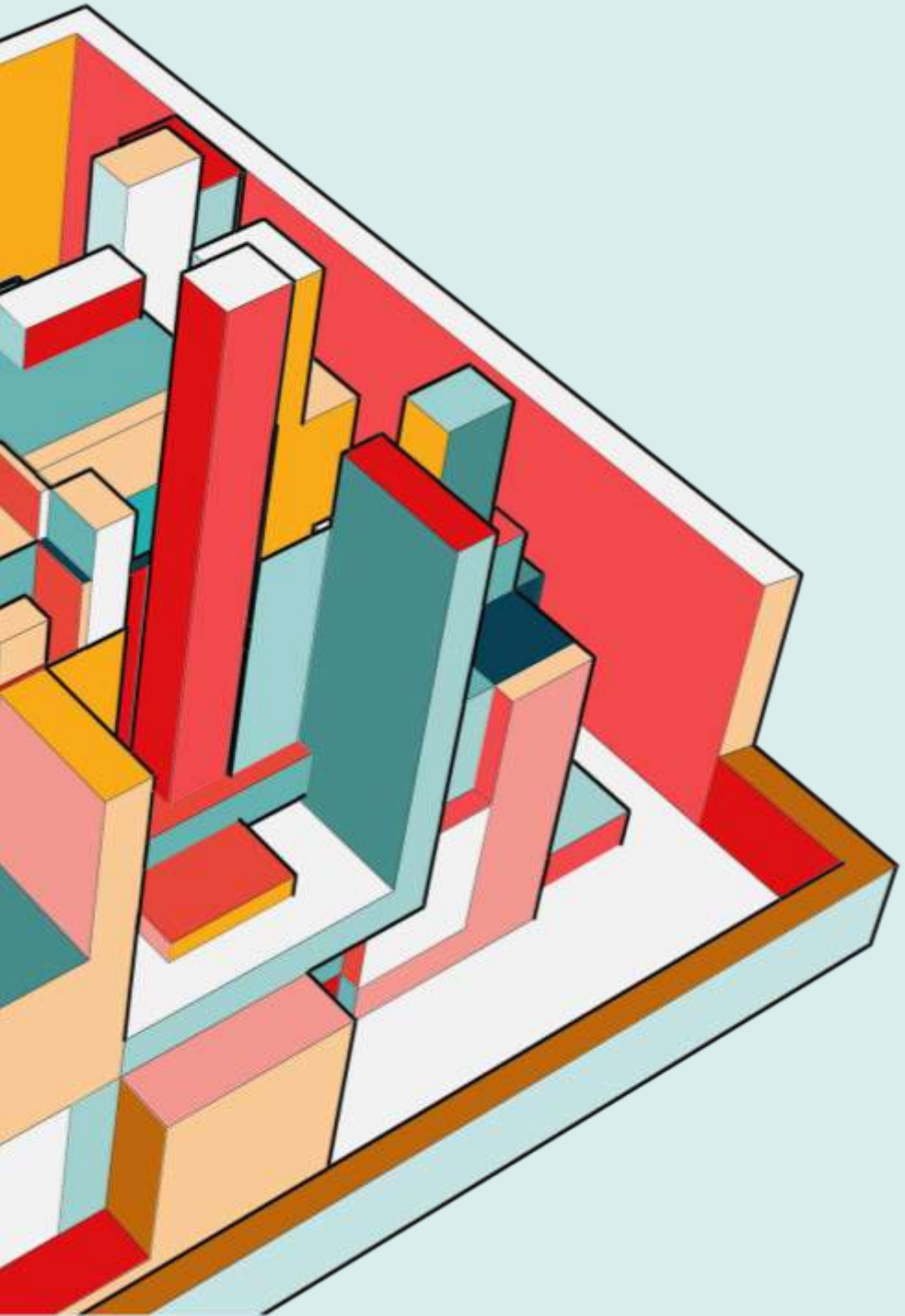
PRACTICAL IMPLEMENTATION

THE PIPELINE



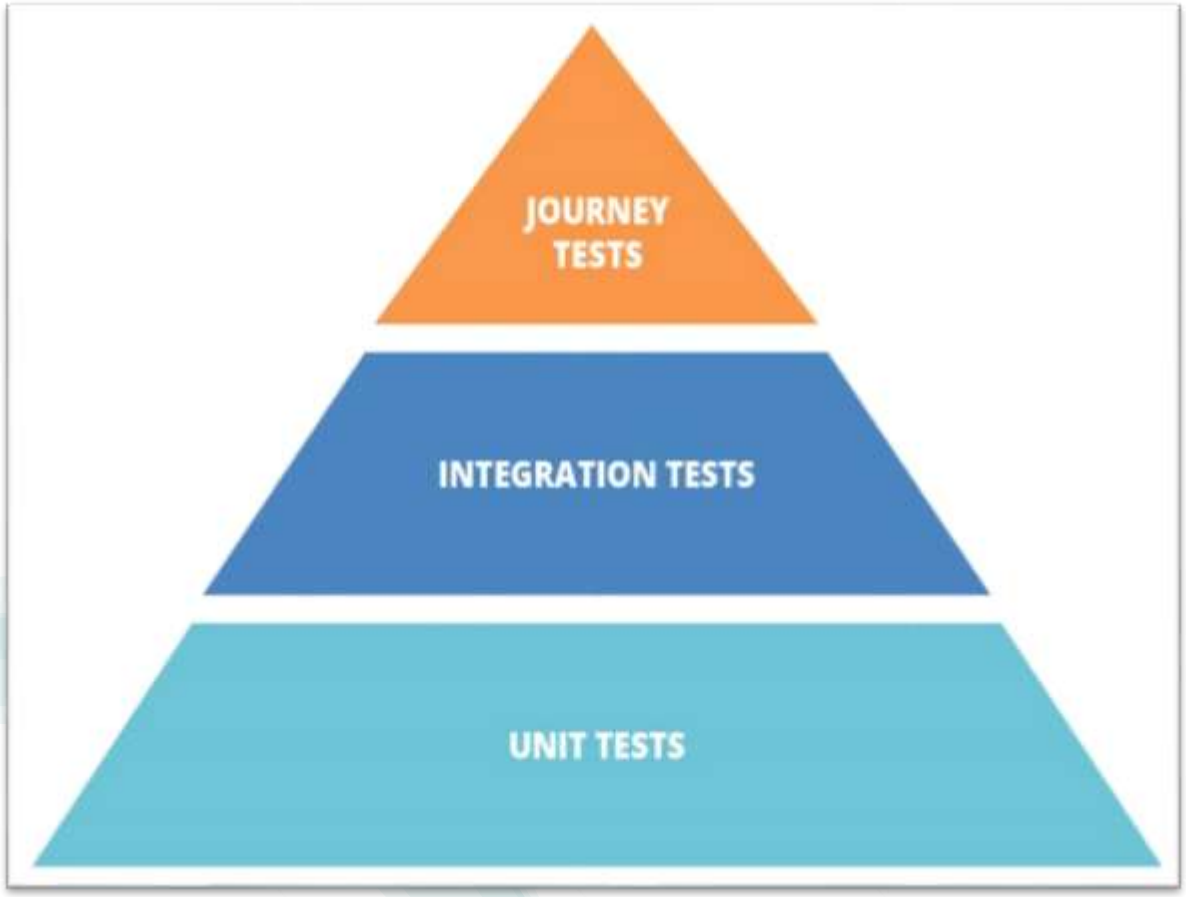
THE PIPELINE



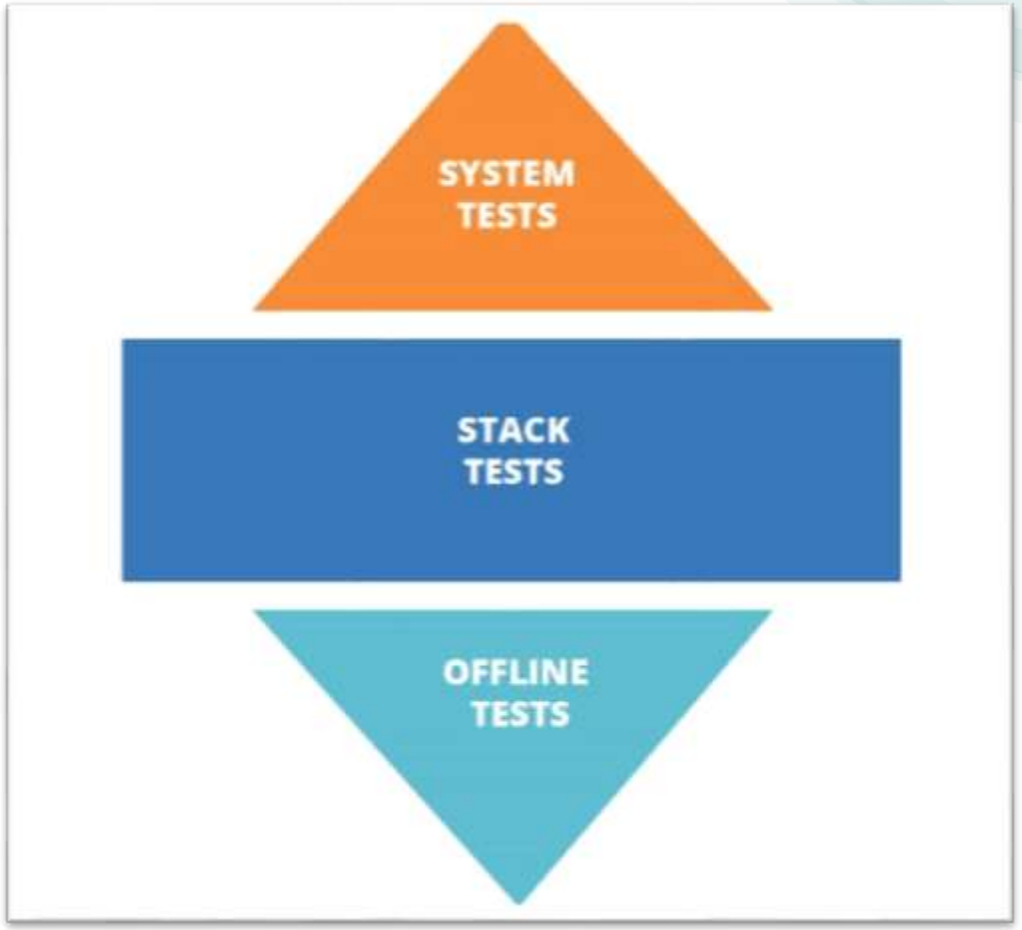


TESTING THE STACK

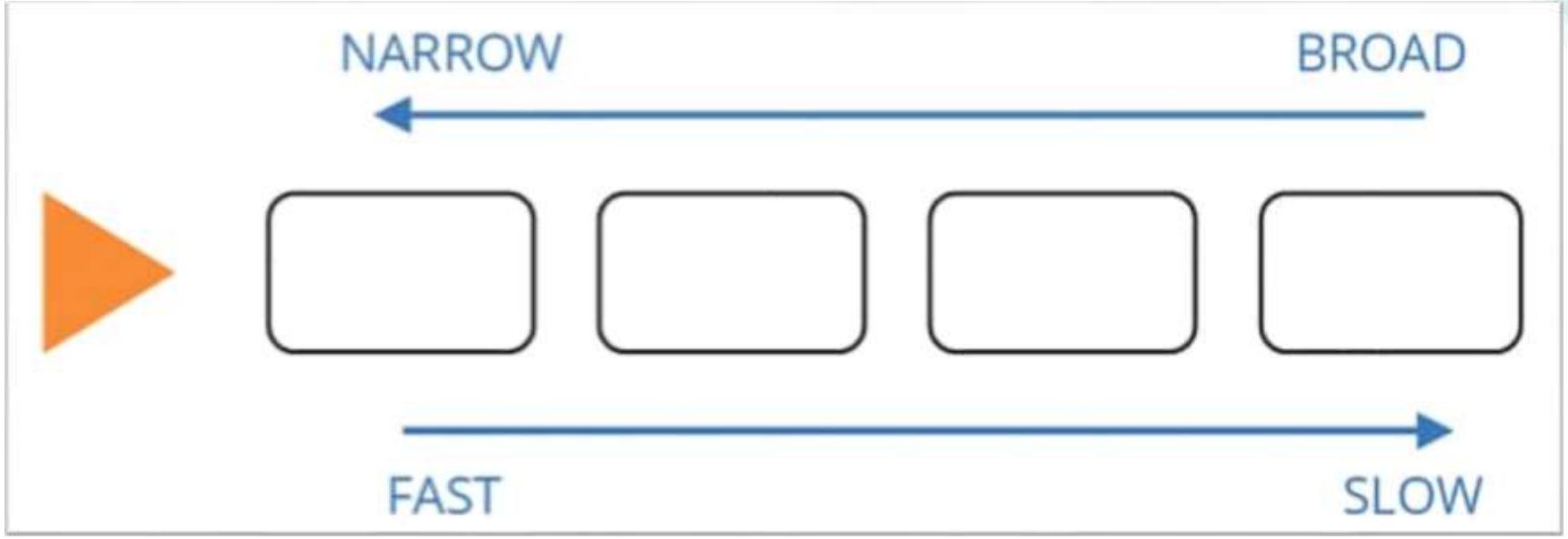
Stages in Infra Testing



Traditional Progressive Testing



IaC Testing Pyramid



PROGRESSIVE TESTING

Start with small, fast tests and progress towards large but slow tests.

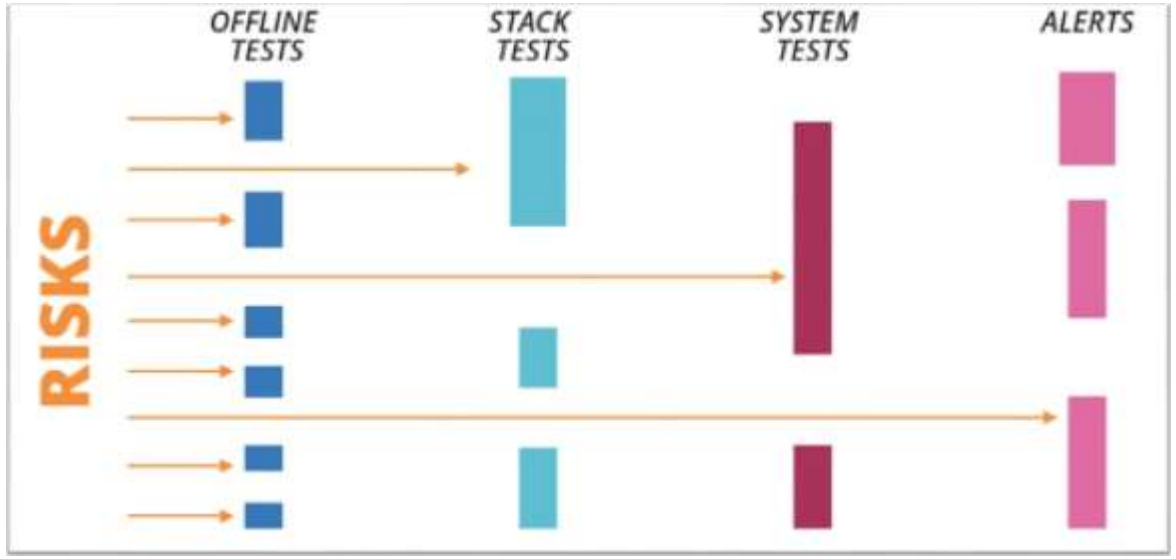
- **Offline Tests:**

Static Analysis, Syntax Checking, security vulnerabilities by scanning code syntax.

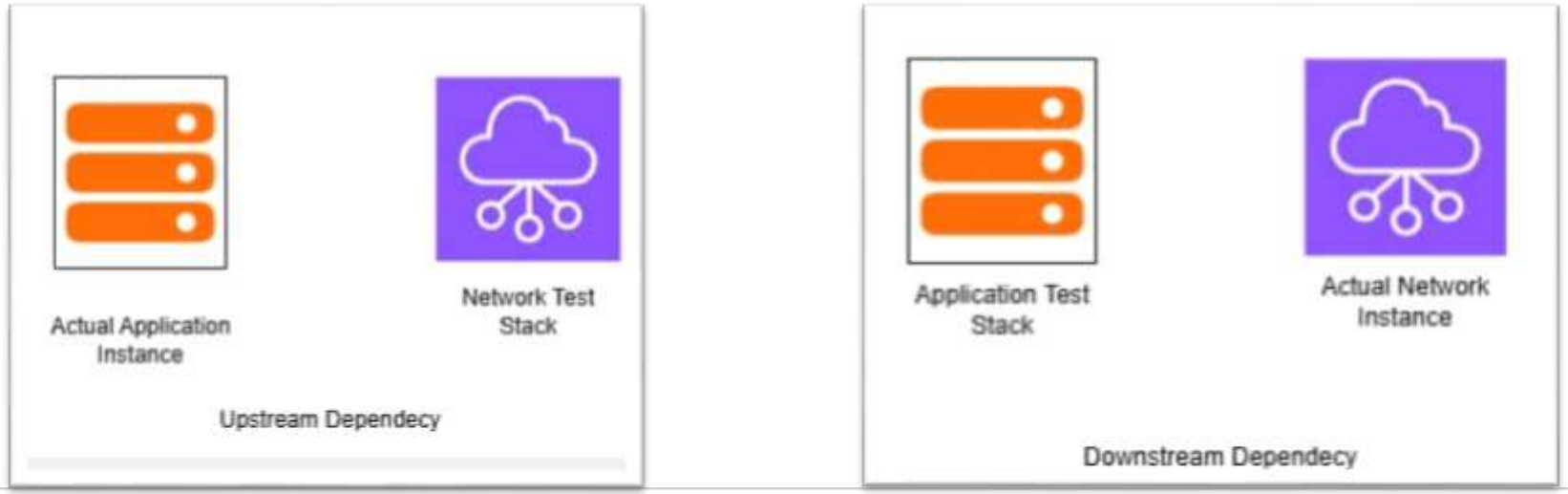
- **Online Tests:**

Stack tests, integration test, known as Online Test

TESTS SHOULD BE CREATED AGAINST RISKS

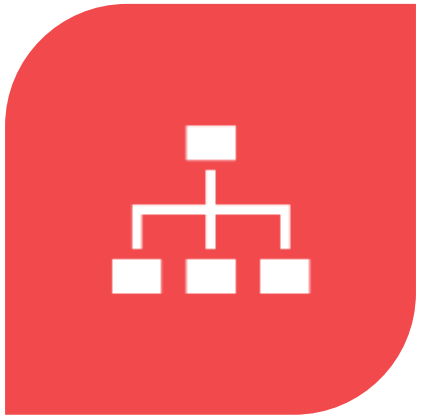


TEST FIXTURES AND TEST DOUBLES



Test Fixture **Test Doubles**

REFACTORING COMPONENTS SO THEY CAN BE ISOLATED



CREATE LOOSELY-COUPLED COMPONENTS.

CREATE TEST WHILE WRITING THE SYSTEM NOT AFTERWARDS.

FOLLOW THE CORE PRINCIPLE OF WRITING **SMALL, SIMPLE PIECES**.

THANK YOU

Let’s discuss this further!!