# Who broke the build?

Using Kuttl to improve E2E testing and release faster

RAM MOHAN RAO CHUKKA | JFROG

Who broke the build?

Manager

Developer

# Who broke the build?

Manager

Developer

# Who broke the build?

# RAM MOHAN RAO CHUKKA

**JFrog**

@ichukka

About me:

Senior Software  Engineer, R&D

India  🇮🇳

Passionate about open source

Love playing table tennis

# JFROG IN A SLIDE

**2008**
Founded

**Public**
NASDAQ Listed

**1100+**
Employees

**9 locations**
7 countries

**7,000+**
Customers

FORTUNE 100 COMPANY

**>70%**
of the
Fortune 100

**6 Products**
from Git to K8S

**All Hybrid**
From OSS to Multi Cloud

**Universal DevOps**
Over 25 tech supported

**Community Champions**

# SOFTWARE UPDATES MATTER



"During the update process you **will not be able** to drive the vehicle"

# JFROG'S MISSION IS TO POWER ALL
# THE SOFTWARE UPDATES IN THE WORLD

JFrog

# FAST ADOPTION BY ENTERPRISES

**100 NEW LOGOS A MONTH**

**Internet & Software**
Google vmware amazon Twitter IBM salesforce ORACLE YAHOO! slack

**Technology & Electronics**
Linked in Snapchat intel CISCO EMC² CHEF Apple box ERICSSON

**Banking & Finance**
BARCLAYS Fidelity INVESTMENTS CHASE citi Bank of America CREDIT SUISSE WELLS FARGO

**Engineering & Aerospace**
ANSYS GM NASA AIRBUS DEFENCE & SPACE BOSCH HALLIBURTON GE BOEING

**Retail & Consumer**
COSTCO WHOLESALE NETFLIX GAP OUTLET Nike VOLVO macy's RIOT GAMES

**Education & Research**
Yale OHIO UNIVERSITY 1804 Stanford University University of San Diego OICR Ontario Institute for Cancer Research MIT

JFrog

# THE
# ECOSYSTEM

aws     Azure     Google Cloud Platform     IBM Cloud

Electric Cloud    puppet    sumologic    ATLASSIAN    MESOSPHERE    WhiteSource    RANCHER

Twistlock    RED HAT OPENSHIFT    NetApp    CHEF    shippable    ORACLE

codefresh    Bamboo    cloudbees    snyk    circleci    Quali    SUSE

pulumi    ca technologies    CLOUD NATIVE COMPUTING FOUNDATION    kubernetes    NeuVector    NUTANIX

bit    Datical    Jenkins    docker    nirmata    ubuntu

JFrog
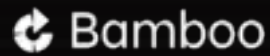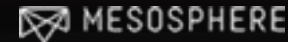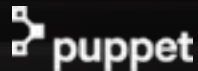
# SOFTWARE DELIVERY HAS EVOLVED



Requirements
Architecture & Design
Development
Quality Assurance
Release
Maintenance

MAINTAIN
REQUIREMENTS
RELEASE
OPS
DESIGN
DEVELOPMENT
TEST

code
plan
build
test
Dev
release
deploy
Ops
operate
monitor
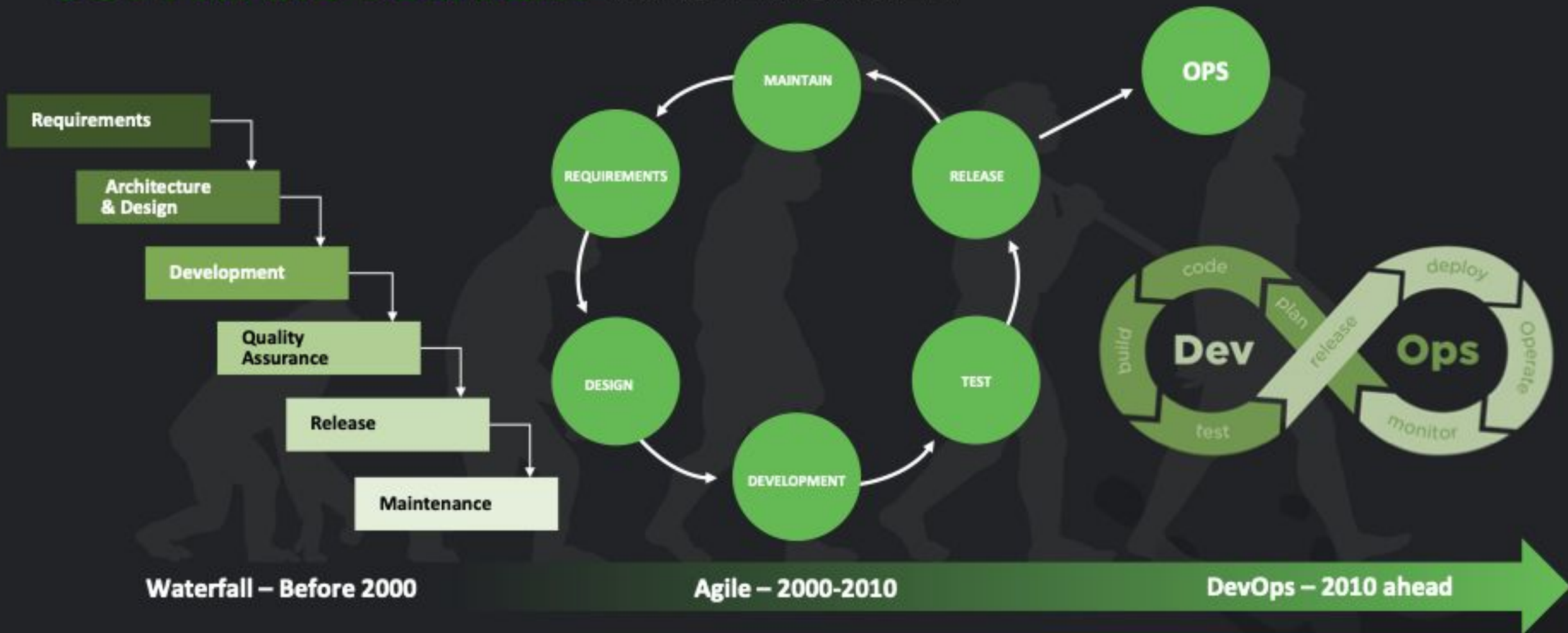
**Waterfall – Before 2000**

**Agile – 2000-2010**

**DevOps – 2010 ahead**

Windows 95
August 1995

First Service Pack
February 1996

Windows 98
June 1998

in amazon spotify NETFLIX
FACEBOOK slack Uber CapitalOne

JFrog
The Liquid Software Company

**Months** Between Releases

**Daily** Releases

5

# Agenda

# Overview

# Ideal development environment



**Single click setup — automation**



**Develop and test locally**



**Same as production environment**

Easy setup

Benefits

Equivalent

# Automation

Generally, setting up a DEV environment for a developer may take a day or two, going through the manual instructions in the wiki page.

It needs to be completely automated and easy to set up.

Easy setup

Benefits

Equivalent

# Automation

Saves time — set up in minutes vs. hours

No manual steps = error free

Quick reload — code-deploy-test cycle

# Equivalent

DEV environment should be same as production environment.

This helps/saves time in reproducing production issues at ease.

# Steps to merge a pull request

Feature development – developer works on new feature

Commit/push – developer writes unit tests based on
the new feature developed

- Tests

- Commits

- Pushes to remote git branch

# Steps to merge a pull request

E2E tests – these tests are performed on a remote CI server.

If tests fail, the developer will:

- Fix → commit

- Push again if tests fail

- Fix and commit again

- Push again

- Round trip continues — how to avoid this ?

Merge/pull request – once E2E tests are successful, the pull request is merged to master branch.

# Remote E2E tests

# Remote E2E tests



Remote

CI/CD server
runs tests

Push

Code development

Unit tests

Pull request

Merge code
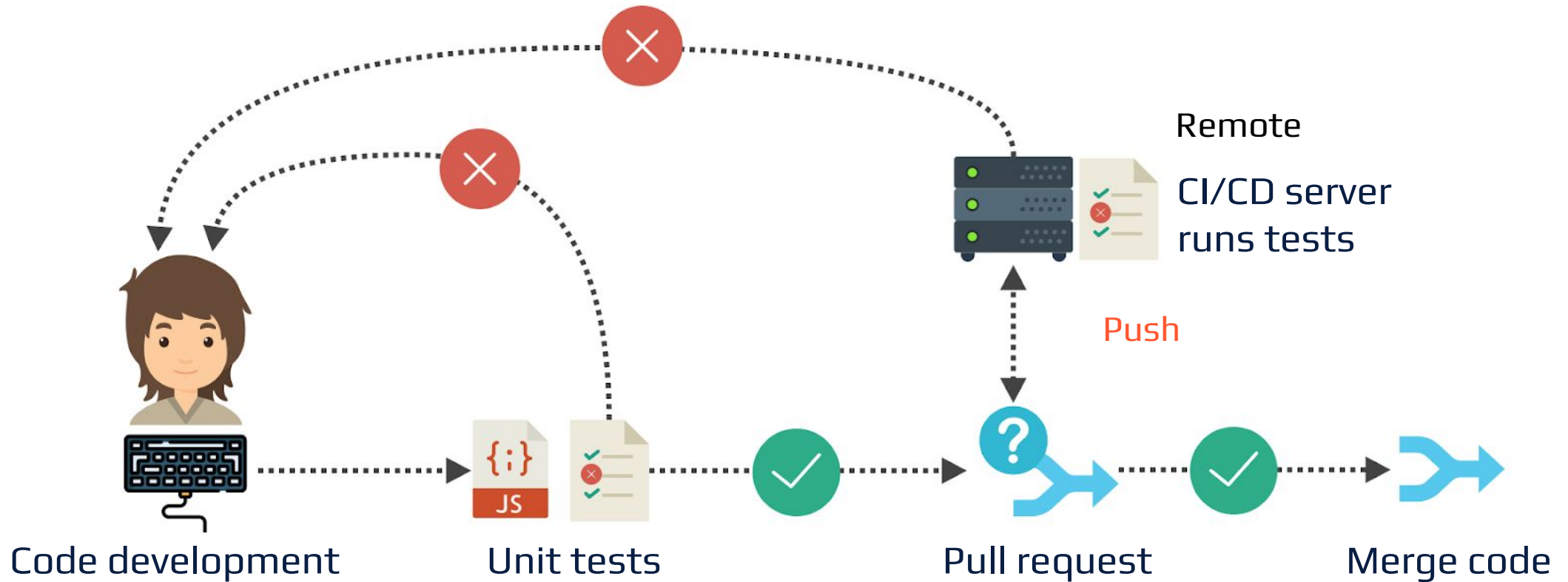
# Local E2E tests

Solution

# Local E2E tests?

Instead of remote E2E tests, use local E2E tests.

Commit local changes to feature branch.

# Local E2E tests?

Instead of committing and pushing to remote server, use git commit --amend.

- If tests fail, fix using git commit --amend.

- If tests fail again, git commit --amend.

This way reduces round trip to fix issues and saves time.

# Local E2E tests



Local

CI/CD server
runs tests

**Amend**

Code development          Unit tests          Pull request          Merge code
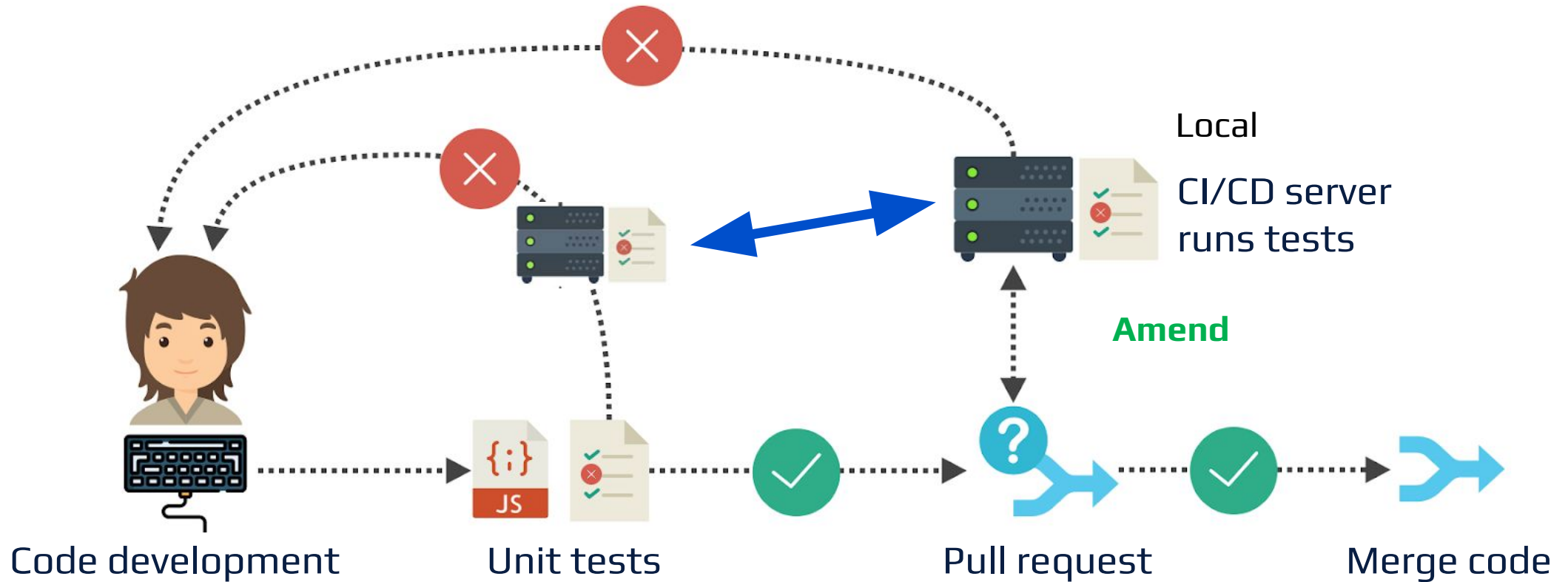
KUTTL

# KUbernetes Test Tool

Tool kit for writing tests, mainly designed for testing operators and controllers, however it can declaratively test any kubernetes objects

YAML-based

Accelerates the ability to create E2E testing in a Kubernetes environment

KUTTL

# Get started

Installation

brew tap kudobuilder/tap && brew install kuttl-cli

kubectl krew install kuttl

go get github.com/kudobuilder/kuttl  #API integration

# KUTTL is for you if:

You are an application admin who wants to automate the creation of a new Kubernetes environment.

You want to test Kubernetes applications on multiple versions of Kubernetes.

You are a developer to wants an easy way to test operators without writing go code.

KUTTL

Example

# Test suite configuration

```
1    apiVersion: kuttl.dev/v1beta1
2    kind: TestSuite
3    startKIND: false
4    name: e2e
5    testDirs:
6      - ./e2e
7    commands:
8      - command: chmod 600 ../tests/kubeconfig
9      - command: helm repo add jfrog https://charts.jfrog.io
10     - command: helm repo update
11   timeout: 300
```

KUTTL

Example

# Test step

```
1    apiVersion: kuttl.dev/v1beta1
2    kind: TestStep
3  ∨ commands:
4      - command: echo "Running default install in $NAMESPACE"
5      - command: helm upgrade --install artifactory --namespace $NAMESPACE jfrog/artifactory
```

KUTTL

Example

# Test step assert

```
1    apiVersion: apps/v1
2    kind: StatefulSet
3    metadata:
4      name: artifactory
5    status:
6      replicas: 1
7      readyReplicas: 1
```

# KUTTL

Example

# Test suite structure

```
∨ demo
  ∨ tests
    ∨ e2e
      ∨ install
        ! 00-assert.yaml
        ! 00-install.yaml
      ∨ scale
        ! 00-assert.yaml
        ! 00-install.yaml
    ! kubeconfig
    ! kuttl-test.yaml
```

```
2    kind: TestStep
3    commands:
4      - command: echo "Running default install in $NAMESPACE"
5      - command: helm upgrade --install artifactory --namespace $NAMESPACE jfrog/artifactory
```

# Demo

# References

KUTTL [Documentation] - https://kuttl.dev/docs/

KUTTL [Github] - https://github.com/kudobuilder/kuttl

Slack Channel [kubernetes] #kudo

https://app.slack.com/client/T09NY5SBT/CG3HTFCMV

K9s [Github] - https://github.com/derailed/k9s

Links

# Summary

Kuttl - Open source Tool

Local E2E Testing

Few broken builds

Release Faster

Happy Developers 🙂

# Questions ?

# Thank you

RAM MOHAN RAO CHUKKA | JFROG