endava

# Elevate Stability :
# & Power Up with E2E Tests!.

endava

# UI Interface

01 /

## Multi-localization

We utilize monorepos.

Each build of every component produces an application for each localization.

02 /

## UI app

It is a sophisticated UI application featuring multiple dashboards and a custom login system.

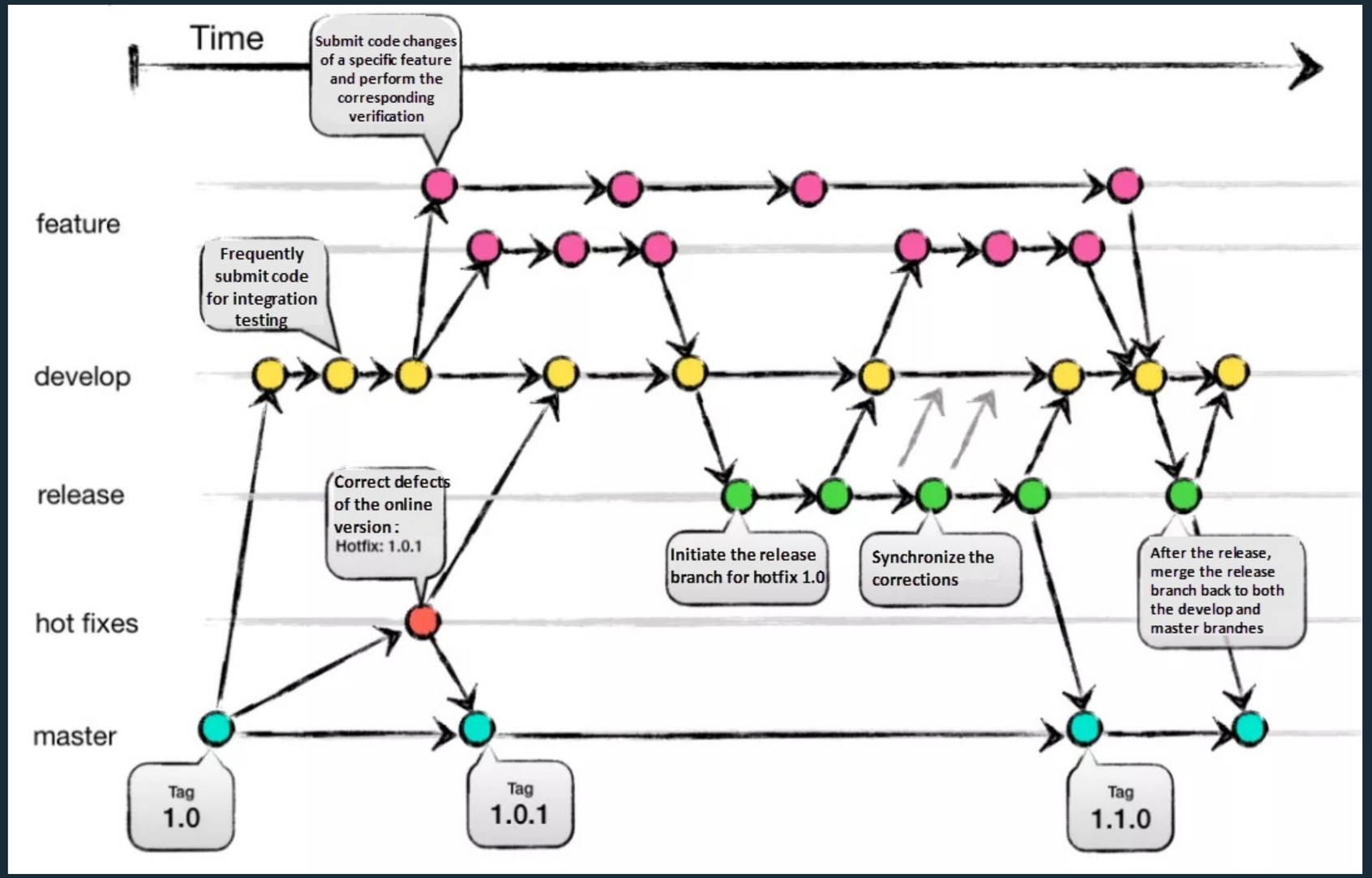
03 /

## Multiple teams

A collaboration of both feature and component teams working on the same project.

endava

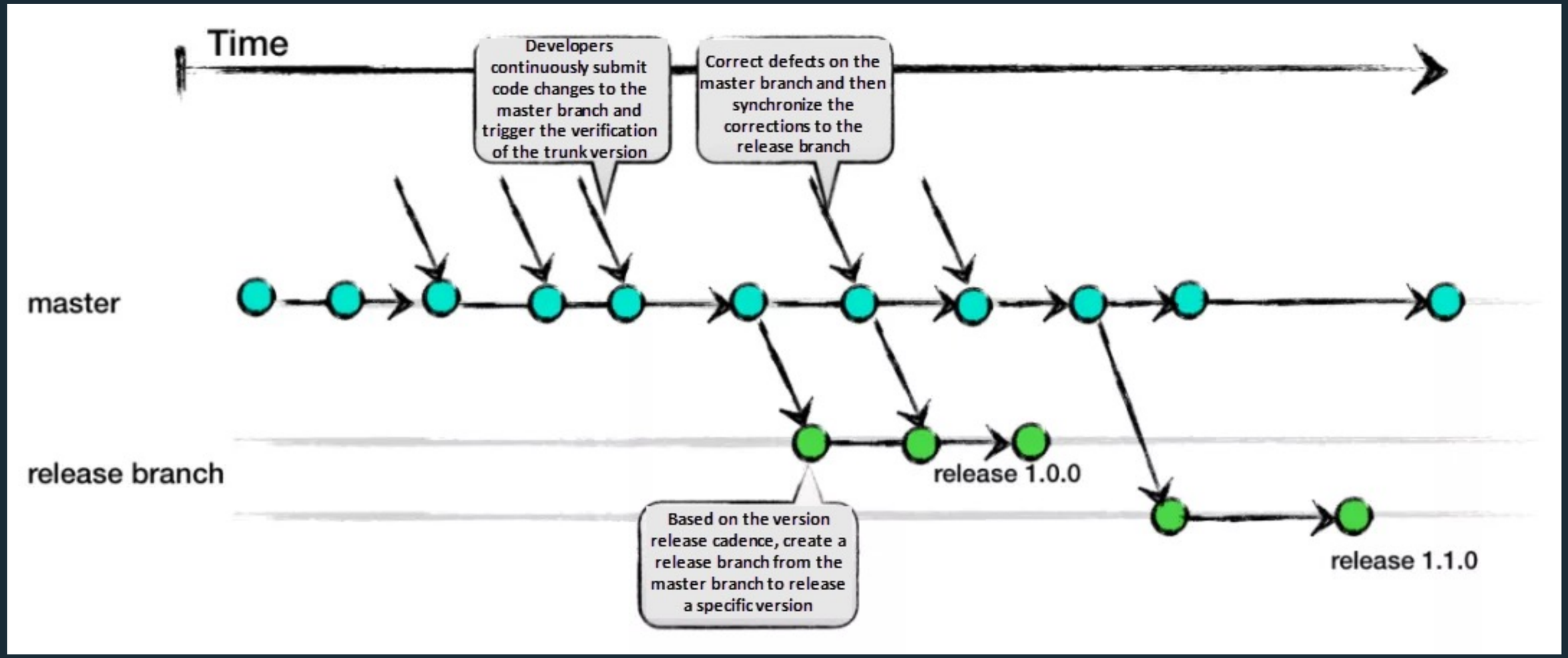
# GIT FLOW




Source: https://medium.com/@sreekanth.thummala/choosing-the-right-git-branching-strategy-a-comparative-analysis-f5e635443423

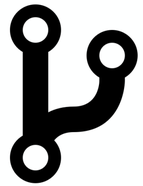
endava

# TRUNK BASED DEVELOPMENT

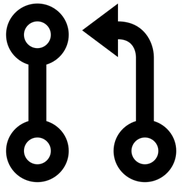



Source: https://medium.com/@sreekanth.thummala/choosing-the-right-git-branching-strategy-a-comparative-analysis-f5e635443423

# Appealing approach

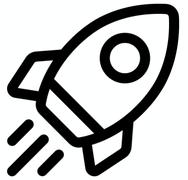


Short-lived branches

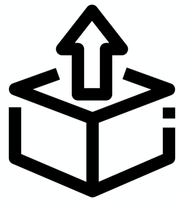


Less merge conflicts

Quick reviews



Main branch is always ready for deploy



Quick releases

endava

We work very fast
 sometimes we may not fully grasp project developments.

# Consequence

As a result, the development team expends capacity on reverting or fixing changes on main branches.

Simultaneously, a failed main branch blocks the testing team from testing new stories, and end-of-day changes impact the night regression results.

# Use Git flow

Give
trunk-based
one more attempt.

# Prerequisites

Feature flags

Good test coverage

Stable CI/CD process

Experienced Team

# We have a plan.

To achieve the desired results, we require these key points:

1. **Create a docker-compose file to launch all our components**
   We opted to consolidate all localized applications, property files, and flyway migrations into a single Docker image. We determine the localization, property file, and whether database migrations need to be executed based on environment variables.
2. **Prepare E2E tests**
   We should take the most crucial tests from each localization's regression pack.
3. **Reconfigure Jenkins agents**
   Generate a temporary environment on demand using Jenkins agents.
4. **Custom GitHub webhooks**
   Configure GitHub webhooks to prevent merges to the main branch that haven't passed the E2E tests.
5. **Custom GitHub comment hooks to trigger E2E tests automatically**
   Initiate E2E tests using the latest builds triggered by the GitHub comment.

# 01 /

## Create a docker-compose file

We create .war and .zip file per localization, but consolidate them into single multi-arch docker image with all .war files, migrations and property files.


Localization-A.war
Localization-B.war
Localization-C.war
Localization-D.war
Localization-E.war

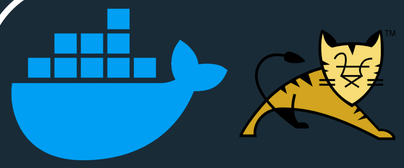
Localization-A-db/
Localization-B-db/
Localization-C-db/
Localization-D-db/
Localization-E-db/

Localization-A.dev.property
Localization-A.sit.property

Localization-B.dev.property

Localization-C.dev.property
Localization-C.sit.property
Localization-C.sit2.property
...

# 01 /



ENTRYPOINT /data/start.py

script runs only once

**Environment variables:**

SPRING_PROFILES_ACTIVE=C (localization)

CONFIG_PATH=/data/config/localization-C.sit.property

RUN_FLYWAY_MIGRATIONS=true

Flyway credentials, schemas and placeholders

If restarted

mv /data/apps/localization-C.war /usr/local/tomcat/webapps/

flyway migrate or flyway clean migrate

touch /data/CONTAINER_ALREADY_STARTED

subprocess.call(['sh', '/usr/local/tomcat/bin/catalina.sh', 'run'])

# 02 /

## Prepare E2E tests

The essential tests should be extracted from each localization regression pack.

Runner Localization-A
Runner Localization-B
Runner Localization-C
Runner Localization-D
Runner Localization-E

Tag E2E

Runner Localization-A-e2e
Runner Localization-B-e2e
Runner Localization-C-e2e
Runner Localization-D-e2e
Runner Localization-E-e2e

Exclude certain tests using the tag: WIP

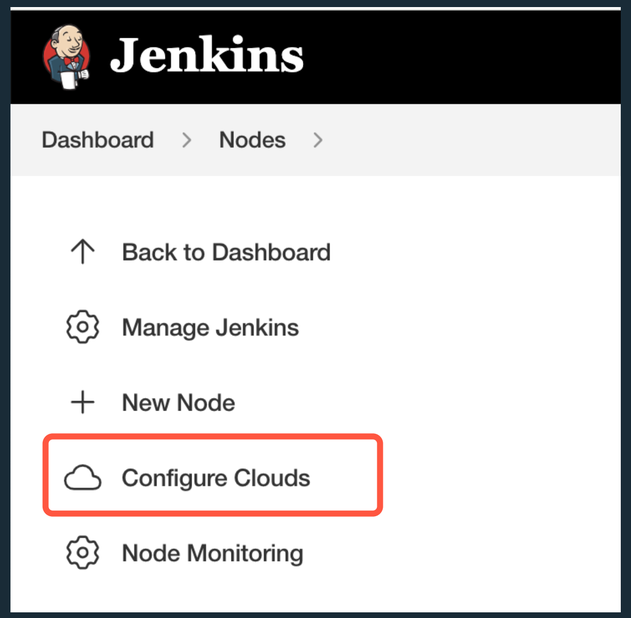
# 03 /

## Reconfigure Jenkins agents

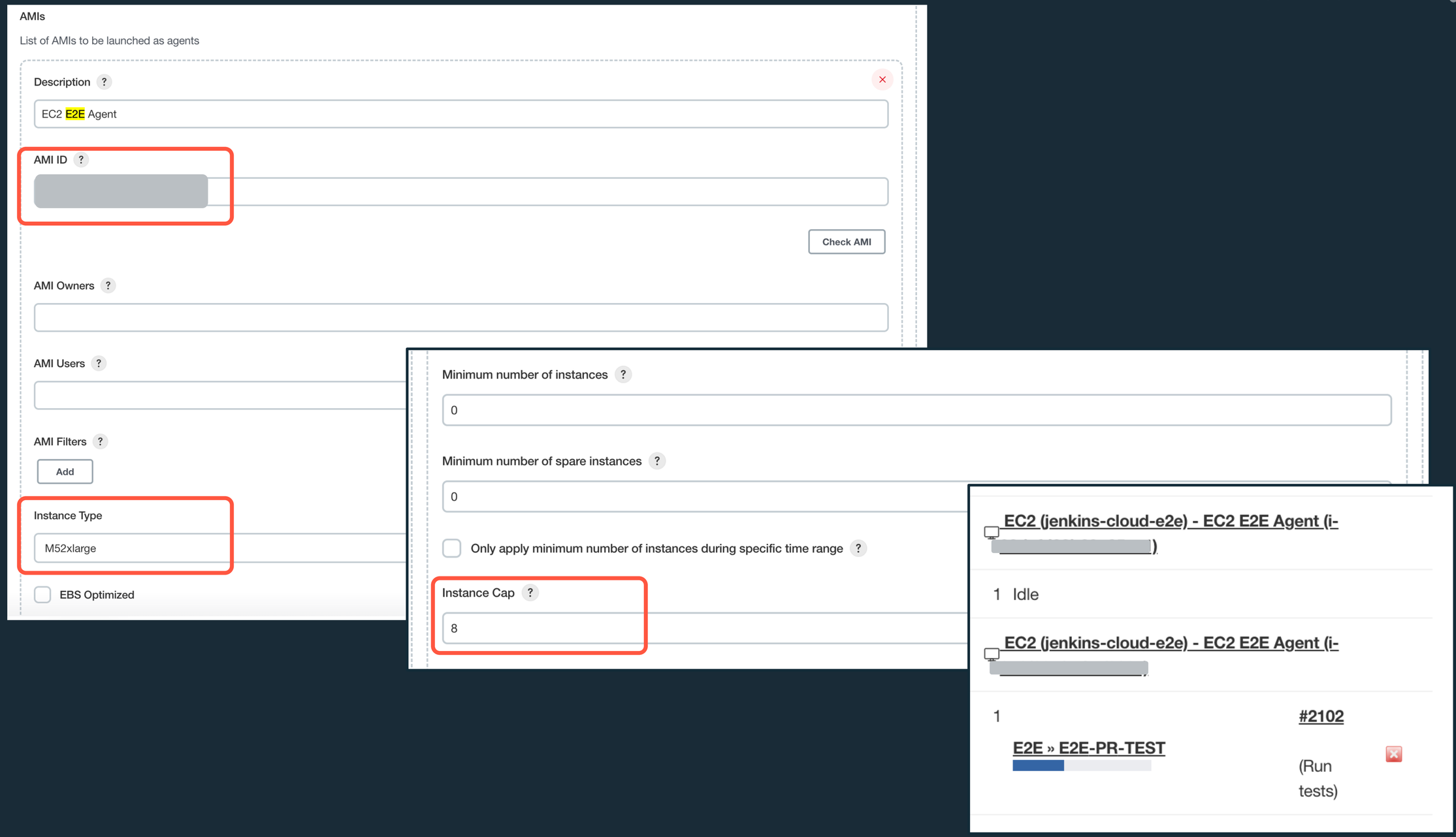
The Jenkins Amazon EC2 plugin can create

Jenkins agents based on the queue.

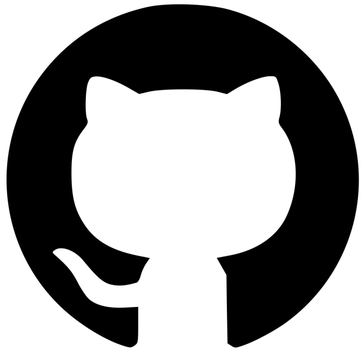
https://plugins.jenkins.io/ec2/



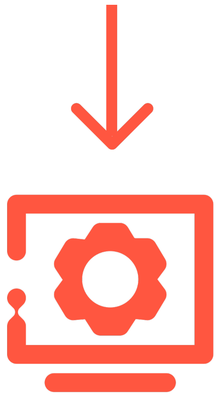

# 03 /



# 04 / Custom GitHub webhooks

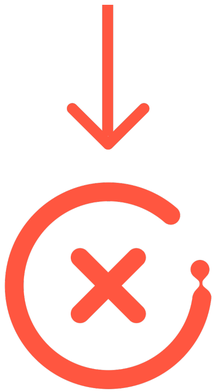


**Jenkinsfile**



Usual CI process

**JenkinsfileE2e**



Refresh E2E status

# 04 /

Jenkins

Search

Dashboard > DevOps > E2E-multibranch-pipelines > [redacted] > Pull Requests (15) > PR-2201 >

- Up
- Status
- Changes
- Build Now
- View Configuration
- Full Stage View
- Job Config History
- Open Blue Ocean
- GitHub
- Pipeline Syntax
- Set Next Build Number

Build History trend

Filter builds...

#16

## Pull Request PR-2201

Full project name: DevOps/E2E-multibranch-pipelines/[redacted]/PR-2201

Recent Changes

## Stage View

| | Declarative: Checkout SCM | Set vars | Check GitHub comment | Run E2E tests |
|---|---|---|---|---|
| Average stage times: | 8s | 3s | 494ms | 384ms |
| #16 Nov 10 15:16 — 1 commit | 3s | 3s | 441ms | 338ms |
| #15 Nov 10 14:46 — 1 commit | 4s | 3s | 533ms | 317ms |
| #14 Nov 10 14:03 — 3 commits | 8s | 3s | 549ms | 384ms |

# 04 /

**Require status checks to pass before merging**

Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

**Require branches to be up to date before merging**

This ensures pull requests targeting a matching branch have been tested with the latest code
effect unless at least one status check is enabled (see below).

Search for status checks in the last week for this repository

Status checks that are required.

continuous-integration/jenkins/pr-merge

E2E Test Successful

**Code owner review required**

Waiting on code owner review from . Learn more about pull request reviews.

Show all reviewers

1 pending reviewer

**No unresolved conversations**

There aren't yet any conversations on this pull request.

View

**Some checks haven't completed yet**

1 pending and 25 successful checks

Hide all checks

- **E2E Test Successful** *Pending — E2E tests should be run succesfully* Required
- **Build, Test, Sonar, Release & Publish ->** — Stage built successfully Details
- **Check GitHub comment** — Stage built successfully Details
- **Clean** — Stage built successfully Details
- **Compile** — Stage built successfully Details
- **Create package** — Stage built successfully Details

**Merging is blocked**

Merging can be performed automatically with 3 approving reviews.

View rules

Merge without waiting for requirements to be met (bypass branch protections)

Rebase and merge

You can also open this in GitHub Desktop or view command line instructions.

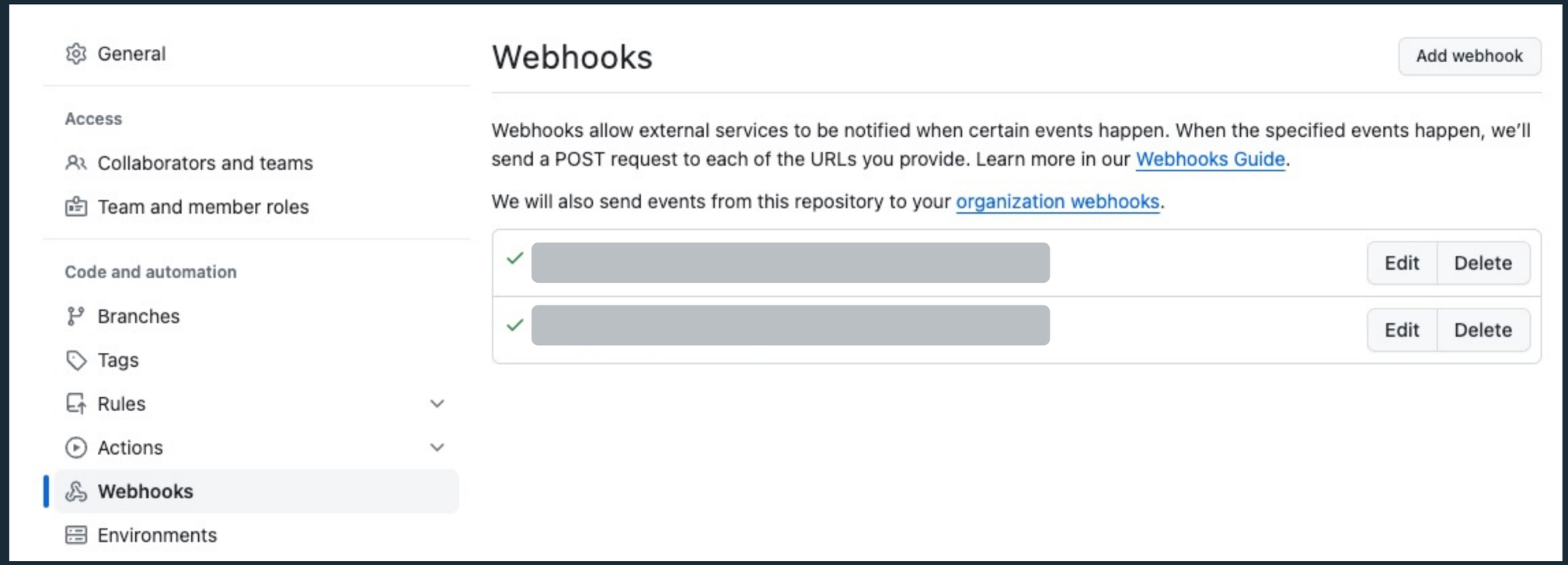
05 /

# Create GitHub comment hooks to start E2E tests automatically

**Jenkins GitHub Pull Request Builder plugin is able to create**

Jenkins agents based on the queue.

https://plugins.jenkins.io/ghprb/



# Webhooks / Manage webhook

Settings | Recent Deliveries

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

**Payload URL** *

https:/[redacted]/ghprbhook/

**Content type**

application/json

**Secret**

**SSL verification**

By default, we verify SSL certificates when delivering payloads.

Enable SSL verification | Disable (not recommended)

- [ ] **Forks**
  Repository forked.
- [ ] **Issues**
  Issue opened, edited, deleted, transferred, pinned, unpinned, closed, reopened, assigned, unassigned, labeled, unlabeled, milestoned, demilestoned, locked, or unlocked.
- [ ] **Merge groups**
  Merge Group requested checks, or was destroyed.
- [ ] **Milestones**
  Milestone created, closed, opened, edited, or deleted.
- [ ] **Page builds**
  Pages site built.
- [ ] **Project columns**
  Project column created, updated, moved or deleted.
- [ ] **Pull request review comments**
  Pull request diff comment created, edited, or deleted.
- [ ] **Pull request reviews**
  Pull request review submitted, edited, or dismissed.
- [x] **Issue comments**
  Issue comment created, edited, or deleted.
- [ ] **Labels**
  Label created, edited or deleted.
- [ ] **Meta**
  This particular hook is deleted.
- [ ] **Packages**
  GitHub Packages published or updated in a repository.
- [ ] **Project cards**
  Project card created, updated, or deleted.
- [ ] **Projects**
  Project created, updated, or deleted.
- [ ] **Pull request review threads**
  A pull request review thread was resolved or unresolved.
- [x] **Pull requests**
  Pull request assigned, auto merge disabled, auto merge enabled, closed, converted to draft, demilestoned, dequeued, edited, enqueued, labeled, locked, milestoned, opened, ready for review, reopened, review request removed, review requested, synchronized, unassigned, unlabeled, or unlocked.

[redacted] / JenkinsfileE2e

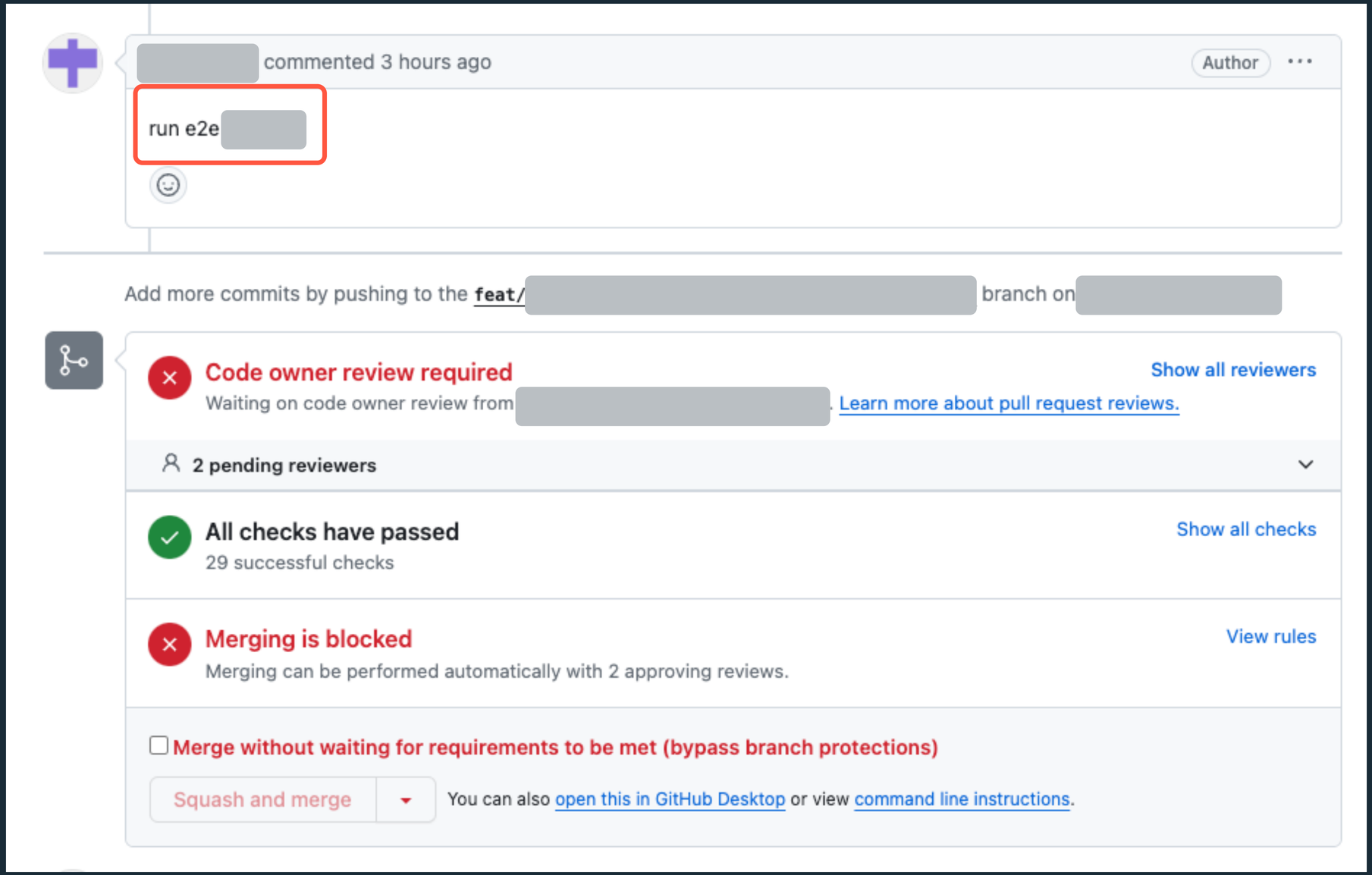
[redacted]

Code | Blame | 47 lines (39 loc) · 934 Bytes

```groovy
@Library('Shared@shared') _

pipeline {
    agent any

    environment {
        [redacted]
    }

    options {
        disableConcurrentBuilds(abortPrevious: true)
        buildDiscarder logRotator(numToKeepStr: '30')
    }

    triggers {
        issueCommentTrigger('.*run e2e.*')
    }
```

```groovy
def call (Map config) {
    [redacted]
    try {
        def comment = currentBuild.rawBuild.getCauses()[0].comment
        def matcher = (comment =~ /.*run e2e ([a-zA-Z0-9-]*).*/)
        region = matcher[0][1]
        println "REGION ${region}"
    } catch (err) {
        resetE2eStatus = "true"
        println "Build has not been triggered by github comment"
    }
    [redacted]
}
```

```groovy
def call(Map config) {
    if (config.resetE2eStatus != "true") {
        buildE2e(REGION: config.REGION)
    } else if (config.createRC != 'true') {
        setStatusPr(state: "pending", e2eRepositories: ["[redacted]": BRANCH])
    }
}
```



Dashboard > E2E > E2E-PR-TEST >

- Up
- Status
- Changes
- Build with Parameters
- Configure
- Delete Pipeline
- Move
- Full Stage View
- Job Config History
- Open Blue Ocean
- Rename
- Cucumber reports (ATF test report)
- Pipeline Syntax
- Set Next Build Number

Build History trend

Filter builds...

#2053
Feb 13, 2024 5:18 PM

# Pipeline E2E-PR-TEST

Full project name: E2E/E2E-PR-TEST

Add description

Disable Project

Recent Changes

## Stage View

| | Declarative: Checkout SCM | Declarative: Tool Install | Clean workspace and set vars | Set PR status for E2E tests | Start Selenium hub | Create docker-compose | Waiting for the healthcheck | Update BPS and IPS mock | Run tests | Generate HTML report | Declarative: Post Actions |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Average stage times: (Average full run time: ~26min 51s) | 29s | 4s | 40s | 3s | 1min 43s | 1min 35s | 8s | 3s | 18min 49s | 8s | 27s |
| #2053 Feb 13 19:18 344 commits | 26s | 5s | 37s | 3s | 2min 0s | 1min 37s | 7s | 3s | 19min 17s | 8s | 25s |
| #2052 Feb 13 16:56 344 commits | 30s | 5s | 42s | 3s | 1min 53s | 1min 33s | 7s | 3s | 20min 16s | 9s | 28s |
| #2051 Feb 13 15:42 343 commits | 26s | 5s | 37s | 3s | 2min 1s | 1min 40s | 7s | 4s | 20min 25s | 10s | 29s |
| #2050 Feb 13 13:01 343 commits | 37s | 5s | 50s | 3s | 2min 11s | 2min 10s | 8s | 4s | 20min 33s | 9s | 27s |
| #2049 Feb 13 11:28 343 commits | 19s | 312ms | 38s | 3s | 17s | 53s | 8s | 3s | 13min 24s | 5s | 27s |

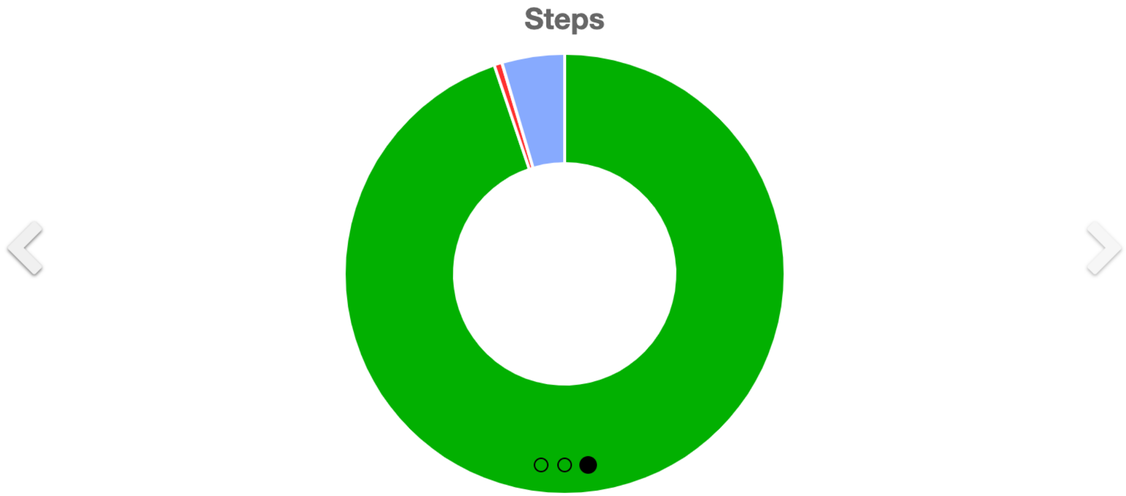
Cucumber Report

Jenkins | Previous results | Latest results | Features | Tags | Steps

| Project | Number | Date |
|---|---|---|
| E2E-PR-TEST | 2049 | 13 Feb 2024, 09:44 |

## Features Statistics

The following graphs show passing and failing statistics for features

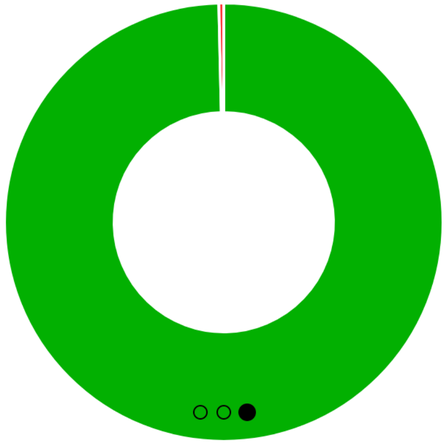



| | Steps | | | | | | Scenarios | | | Features | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Feature | Passed | Failed | Skipped | Pending | Undefined | Total | Passed | Failed | Total | Duration | Status |
| [redacted] | 70 | 0 | 0 | 0 | 0 | 70 | 8 | 0 | 8 | 3:29.938 | Passed |
| | 17 | 0 | 0 | 0 | 0 | 17 | 1 | 0 | 1 | 25.654 | Passed |
| | 78 | 1 | 8 | 0 | 0 | 87 | 7 | 1 | 8 | 5:7.911 | Failed |
| | **165** | **1** | **8** | **0** | **0** | **174** | **16** | **1** | **17** | **9:3.504** | **3** |
| | **94.83%** | **0.57%** | **4.60%** | **0.00%** | **0.00%** | | **94.12%** | **5.88%** | | | **66.67%** |

Cucumber Report



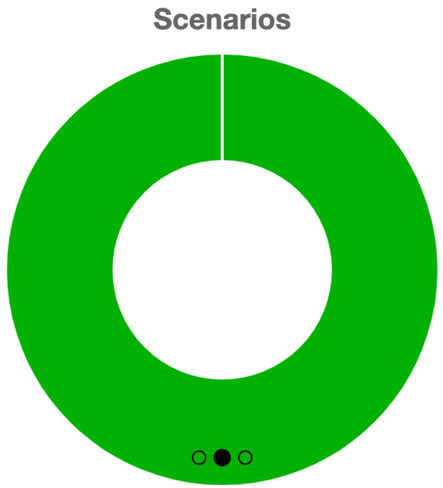
## Steps



| | Steps | | | | | | Scenarios | | | Features | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Feature | Passed | Failed | Skipped | Pending | Undefined | Total | Passed | Failed | Total | Duration | Status |
| [redacted] | 3 | 0 | 0 | 0 | 0 | 3 | 1 | 0 | 1 | 11.623 | Passed |
| | 12 | 0 | 0 | 0 | 0 | 12 | 1 | 0 | 1 | 30.508 | Passed |
| | 22 | 0 | 0 | 0 | 0 | 22 | 1 | 0 | 1 | 1:10.153 | Passed |
| | 13 | 1 | 0 | 0 | 0 | 14 | 0 | 1 | 1 | 1:28.951 | Failed |
| | 84 | 0 | 0 | 0 | 0 | 84 | 1 | 0 | 1 | 3:19.106 | Passed |
| | 20 | 0 | 0 | 0 | 0 | 20 | 1 | 0 | 1 | 2:4.828 | Passed |
| | 44 | 0 | 0 | 0 | 0 | 44 | 1 | 0 | 1 | 1:47.557 | Passed |
| | 12 | 0 | 0 | 0 | 0 | 12 | 1 | 0 | 1 | 17.476 | Passed |
| | 13 | 0 | 0 | 0 | 0 | 13 | 1 | 0 | 1 | 32.929 | Passed |
| | 19 | 0 | 0 | 0 | 0 | 19 | 1 | 0 | 1 | 28.502 | Passed |
| | 38 | 0 | 0 | 0 | 0 | 38 | 1 | 0 | 1 | 1:42.973 | Passed |
| | **280** | **1** | **0** | **0** | **0** | **281** | **10** | **1** | **11** | **13:34.609** | **11** |
| | **99.64%** | **0.36%** | **0.00%** | **0.00%** | **0.00%** | | **90.91%** | **9.09%** | | | **90.91%** |

# Features Statistics

The following graphs show passing and failing statistics for features




| | Steps | | | | | | Scenarios | | | Features | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Feature | Passed | Failed | Skipped | Pending | Undefined | Total | Passed | Failed | Total | Duration | Status |
| [redacted] | 12 | 0 | 0 | 0 | 0 | 12 | 1 | 0 | 1 | 43.060 | Passed |
| | 32 | 0 | 0 | 0 | 0 | 32 | 1 | 0 | 1 | 1:28.531 | Passed |
| | 89 | 0 | 0 | 0 | 0 | 89 | 1 | 0 | 1 | 3:29.105 | Passed |
| | 14 | 0 | 0 | 0 | 0 | 14 | 1 | 0 | 1 | 1:33.164 | Passed |
| | 20 | 0 | 0 | 0 | 0 | 20 | 1 | 0 | 1 | 2:5.615 | Passed |
| | 44 | 0 | 0 | 0 | 0 | 44 | 1 | 0 | 1 | 1:48.401 | Passed |
| | 12 | 0 | 0 | 0 | 0 | 12 | 1 | 0 | 1 | 18.372 | Passed |
| | 13 | 0 | 0 | 0 | 0 | 13 | 1 | 0 | 1 | 38.179 | Passed |
| | 38 | 0 | 0 | 0 | 0 | 38 | 1 | 0 | 1 | 1:44.097 | Passed |
| | **274** | **0** | **0** | **0** | **0** | **274** | **9** | **0** | **9** | **13:48.528** | **9** |
| | **100.00%** | **0.00%** | **0.00%** | **0.00%** | **0.00%** | | **100.00%** | **0.00%** | | | **100.00%** |

[redacted] 2/5 5:51 PM Edited

## Regression Results on 5 February 2024

Hi Testing! Regression results:

[redacted]: 99.6 % (1 tests failed / 225)
99.7 % (2 tests failed / 654)
[redacted] 94.5 % (44 tests failed / 799)
: 97.8 % (13 tests failed / 763)

👍 1 ❤️ 1

Reply

[redacted] 2/6 6:12 PM

## Regression Results on 6 February 2024

Hi Testing! Regression results:

[redacted]: 99.6 % (1 tests failed / 225)
99.7 % (2 tests failed / 654)
[redacted] 97.9 % (17 tests failed / 799)
98.7 % (9 tests failed / 763)

👍 2

Reply

[redacted] Wednesday 6:38 PM

## Regression Results on 7th February 2024

Hi Testing! Regression results:

[redacted]: 99.1 % (2 tests failed / 225)
99.2 % (5 tests failed / 654)
[redacted] 98.3 % (14 tests failed / 800)
: 98.2 % (13 tests failed / 763)
[redacted]

see more

❤️ 1

Reply

[redacted] Thursday 6:04 PM

## Regression Results on 8th February 2024

Hi Testing! Regression results:

[redacted]: 99.6 % (1 tests failed / 225)
99.7 % (2 tests failed / 654)
[redacted] 98.6 % (11 tests failed / 800)
: 98.8 % (8 tests failed / 763)
[redacted]

see more

❤️ 1

Reply

[redacted] Friday 6:39 PM

## Regression Results on 9th February 2024

[redacted]: 100 % (0 tests failed / 225)
99.2 % (5 tests failed / 654)
[redacted] 98.0 % (16 tests failed / 800)
98.4 % (11 tests failed / 763)

❤️ 1

Reply

# Thank you!


endava