Implementation of Continuous Integration and Continuous Delivery

Based on Jenkins CI

Members in Team

For example our team includes 50 members (developers, QA's). They are developing more than one component for the one huge system. For clear work process management we split them to several separate teams.
In each team will be ~12 people

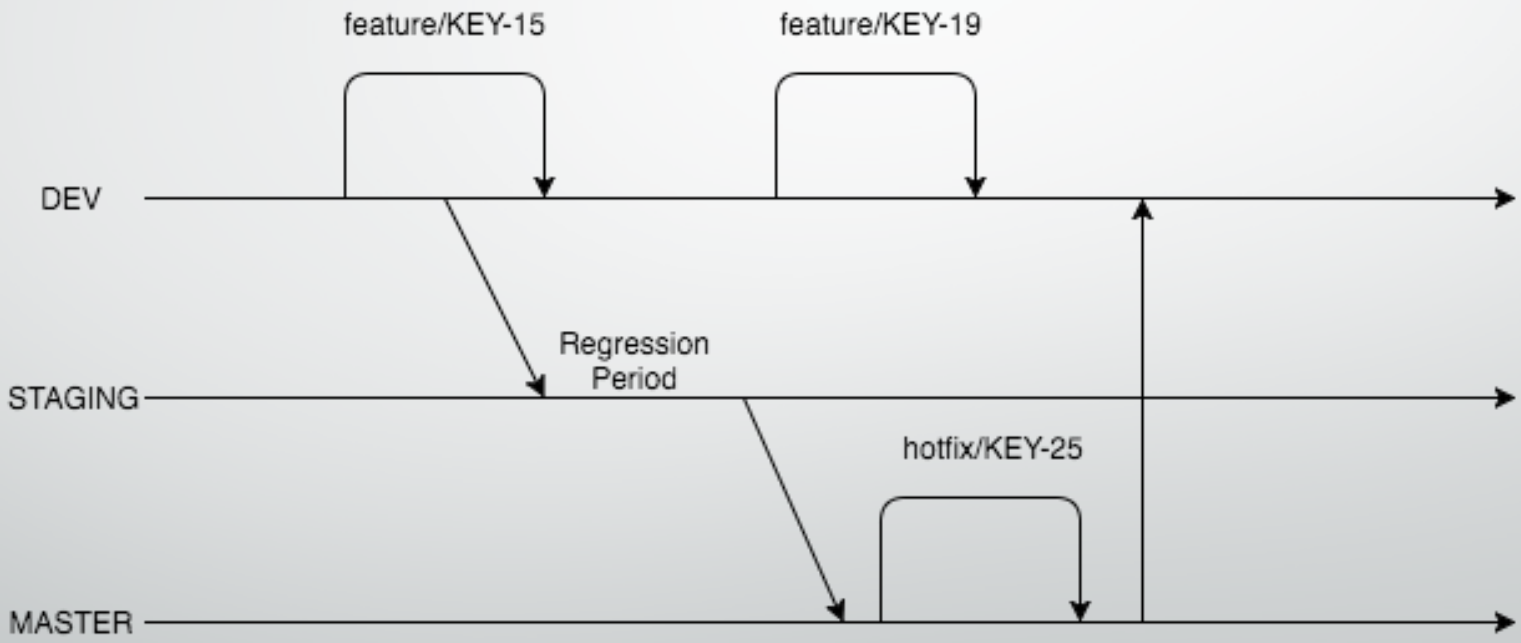
Team representation

Let's imagine that we have 4 teams (50/12 members)

- Frontend team
- Backend team
- iOS team
- Android team

It’s give us more flexible and clear development process

Branching Strategy



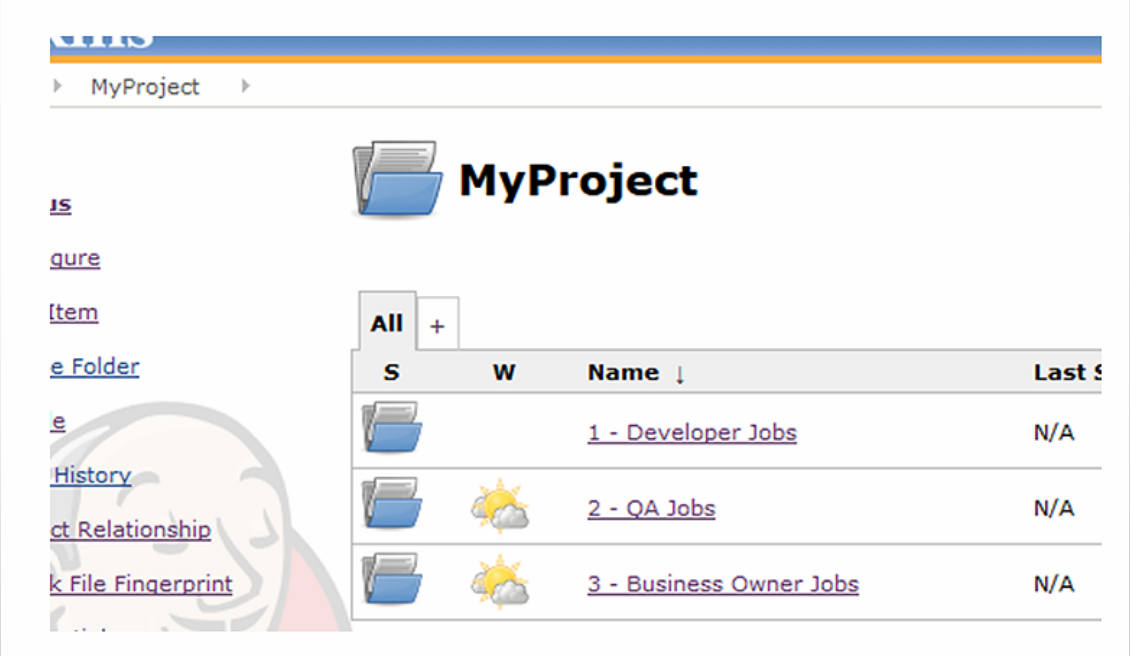
Branching Strategy

Branching strategy released in each team and required to follow next rules:
- branch ‘dev’ assigned for DEV and QA environments integrations between teams
- branch ‘staging’ will be released on Staging for regression tests and pre-production integrations
- ‘master’ branch – final code version for release.

View in Jenkins

Jobs related for each team will be in own folder. Access on it will be based on permissions security.

This will give us the opportunity avoid confusion in future



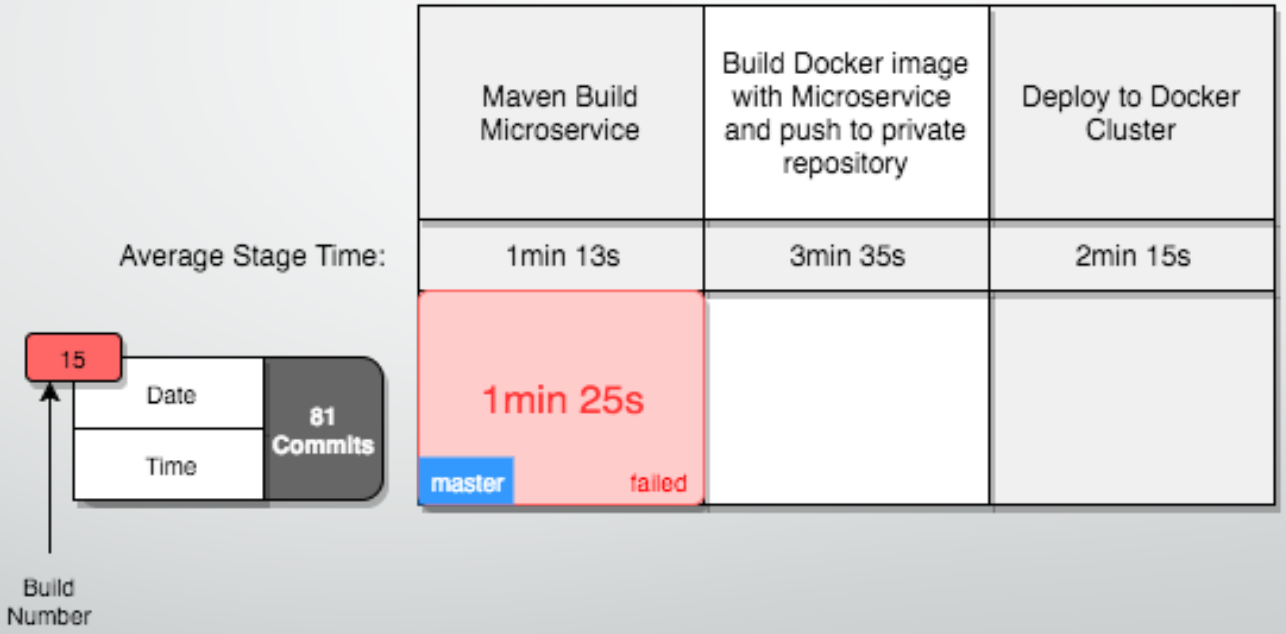
Split jobs by action

We doesn't want to make the build job and the deploy job as one long-run job.
cons of the build and deploy as one job:

- hard debug
- a lot of configuration
- refuse flexibility

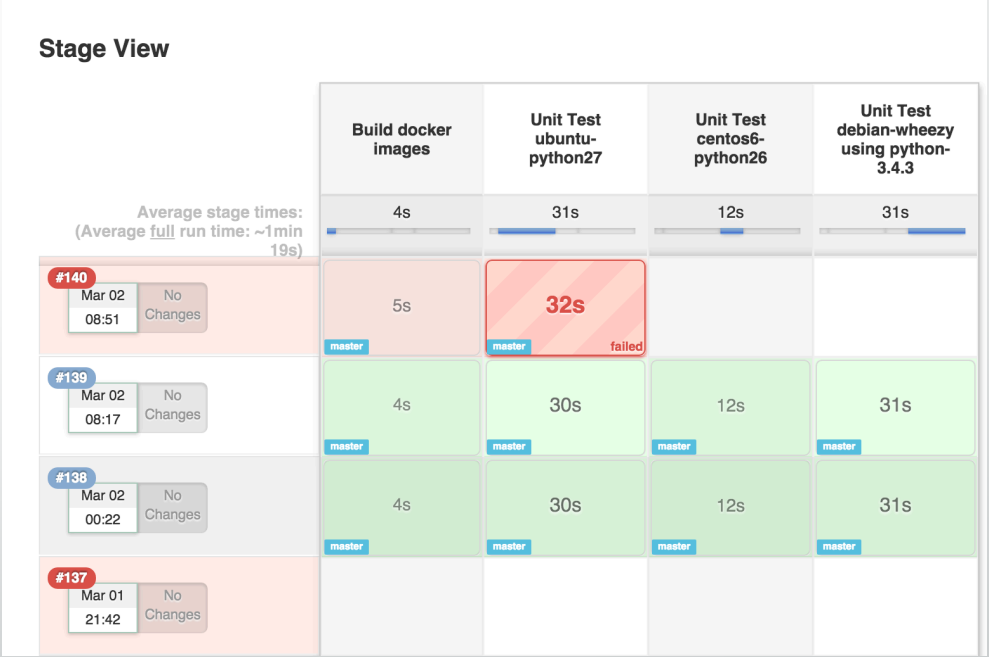
That’s why all Jobs will be separate by action and if build was successful - post-build action will trigger another job, for example deploy to server or create Docker image

Example pipeline view



Pipeline View

You can easily view progress of execution and detect which part is failed



Naming

Name of each pipeline will be same as environment
Dev, QA, Staging, Production

Debug

Easily debug with the console output to find root cause of failed build

Console Output

```
Started by an SCM change
Building in workspace /var/lib/jenkins/jobs/CodeDeployApp/workspace
 > git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://git-codecommit.us-east-1.amazonaws.com/v1/repos/DemoRepository #
timeout=10
Fetching upstream changes from https://git-codecommit.us-east-1.amazonaws.com/v1/repos/DemoRepository
 > git --version # timeout=10
 > git -c core.askpass=true fetch --tags --progress https://git-codecommit.us-east-1.amazonaws.com/v1/repos
/DemoRepository +refs/heads/*:refs/remotes/origin/*
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 40df63b0d7dd567529b677a3cab98495621369d3 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 40df63b0d7dd567529b677a3cab98495621369d3
First time build. Skipping changelog.
Zipping files into /tmp/CodeDeployApp-4674747006653958298.zip
Uploading zip to s3://jenkinscodedeploy-codedeploybucket-9m3kh94ahrbx/CodeDeployApp-4674747006653958298.zip
Registering revision for application 'JenkinsCodeDeploy-DemoApplication-1641E0MP53TNU'
Creating deployment with revision at {RevisionType: S3,S3Location: {Bucket: jenkinscodedeploy-codedeploybucket-
9m3kh94ahrbx,Key: CodeDeployApp-4674747006653958298.zip,BundleType: zip,ETag: 42c8facab37c3732c5700aa4392fe05d},}
Finished: SUCCESS
```

Integrations

Thank you for viewing

Author: Dmitriy Shamenko