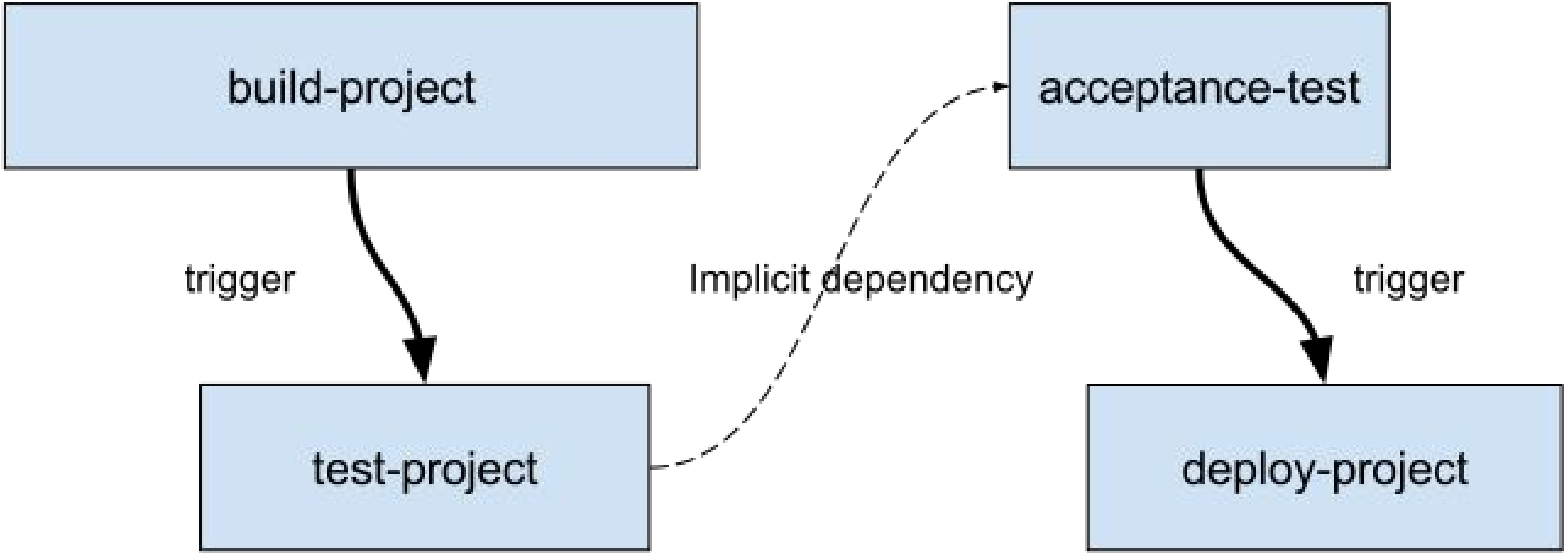
Converting "Freestyle pipelines" to real Pipelines.

The Problem



Implicit/non-obvious "pipeline" of jobs

The Solution

Iteration #0: Freestylin' Pipeline

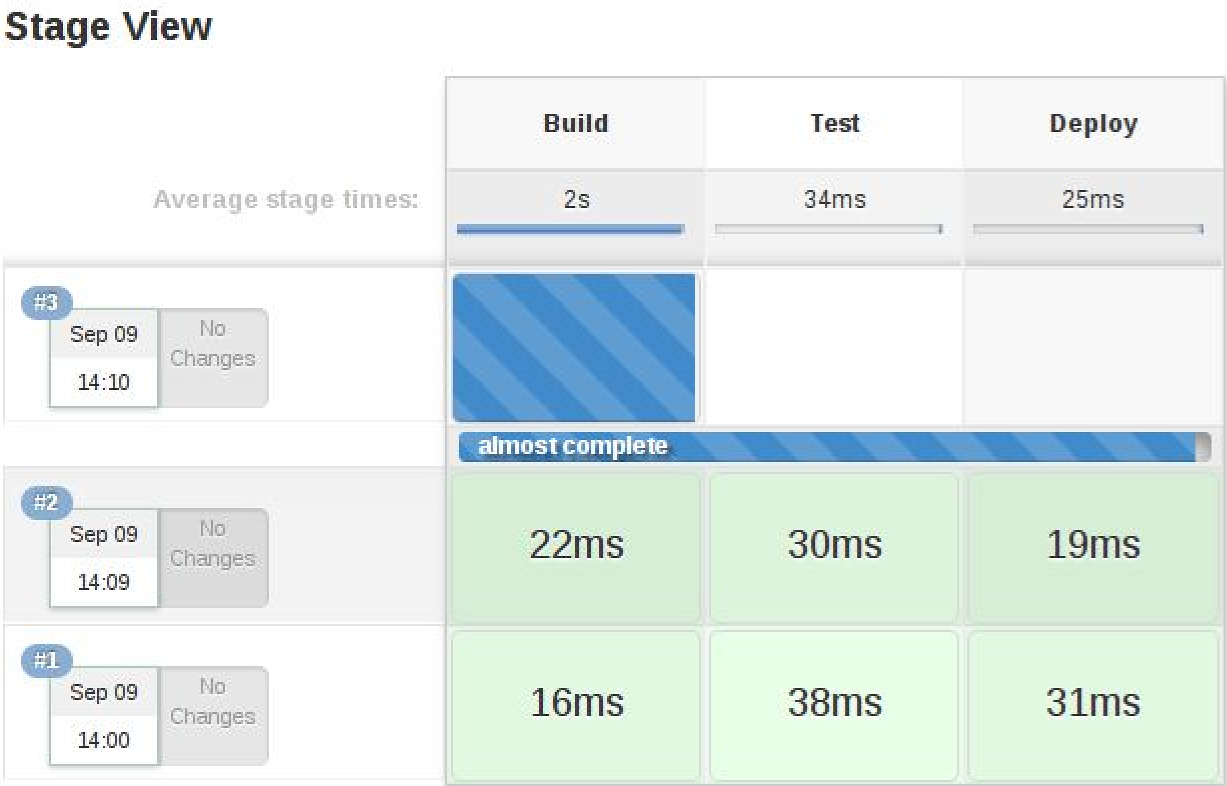
```
node {

  def b = build job: 'build-project'

  build job: 'test-project', parameters: [string(name: 'Upstream', value: b.number)]

  build job: 'deploy-project', parameters: [string(name: 'Upstream', value: b.number)]

}
```

```
git add Jenkinsfile
git commit -m "Pipeline as code!"
```

Iteration #1: Adding stages to the Pipeline

```
  stage 'Build' {
     def b = build job: 'build-project'
  }
  stage 'Test' {
     build job: 'test-project',  parameters: [string(name: 'Upstream', value: b.number)]
  }
  stage 'Deploy' {
     build job: 'deploy-project', parameters: [string(name: 'Upstream', value:
b.number)]
  }
```

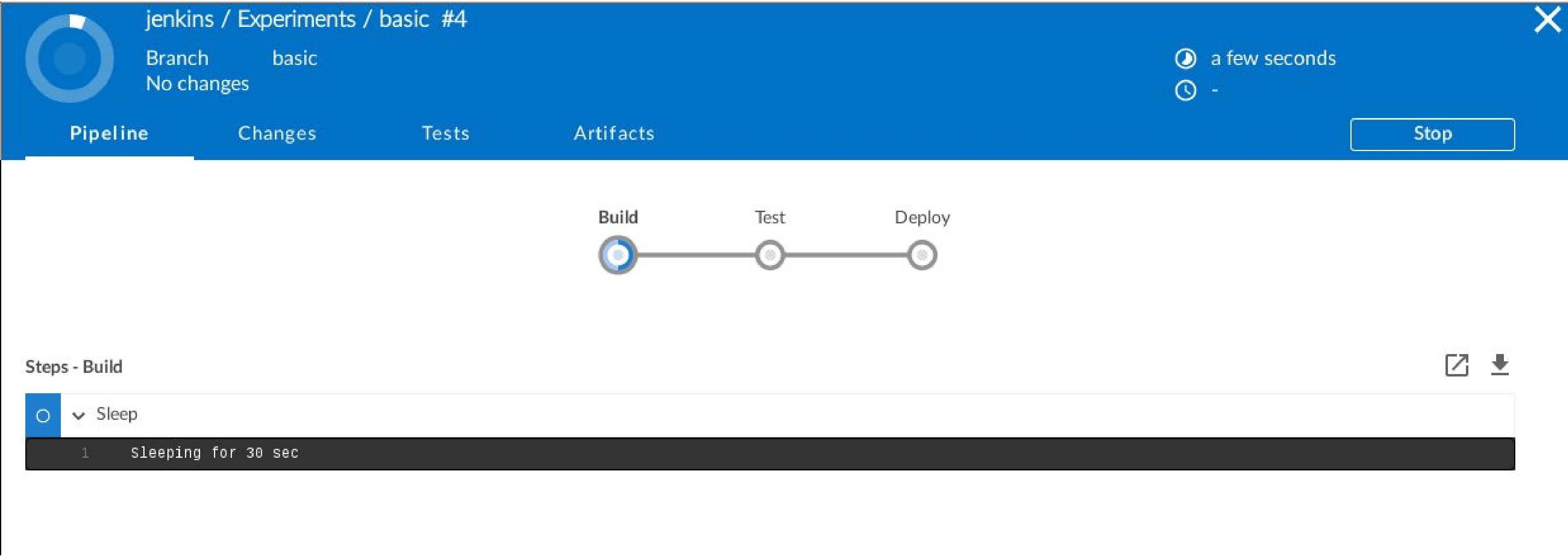
```
git commit Jenkinsfile -m "Visualize!"
```



Stage View

	Build	Test	Deploy
Average stage times:	2s	34ms	25ms
#3 Sep 09 14:10 No Changes			
	almost complete		
#2 Sep 09 14:09 No Changes	22ms	30ms	19ms
#1 Sep 09 14:00 No Changes	16ms	38ms	31ms

Freestyle Pipeline with a centralized view



jenkins / Experiments / basic #4

Branch basic

No changes

a few seconds

-

Steps - Build

○ Sleep

```
Sleeping for 30 sec
```

Freestyle Pipeline with a centralized view

Iteration #2: Going all in on Pipeline

```
stage 'Build' {
   checkout scm
   docker.image('maven:3-jdk-8').inside {
     sh 'mvn clean install'
     stash includes: 'target/**/*.jar', name: 'app'
     archiveArtifacts artifacts: 'target/**/*.jar'
     junit 'target/surefire-reports/**/*.xml'
   }
}
```

Iteration #2: Going all in on Pipeline

```
stage 'Test' {
   unstash 'app'
   git 'git://github.com/jenkins-inc/acceptance-tests.git'
   docker.image('openjdk:8-jre').inside {
     sh './execute-test-harness.sh'
     junit 'features/reports/**/*.xml'
   }
}
```

Iteration #2: Going all in on Pipeline

```
stage 'Deploy' {
   checkout scm
   unstash 'app'
   def container = docker.build("jenkins-inc/nextbigthing:${env.BUILD_ID}")
   /* .. assert container is good .. */
   container.push()
}
```

```
git commit Jenkinsfile \
        -m "The full Pipeline as code!"
```

thank you