

Continuous Integration

Automation with Roundtable TSMS and Jenkins

Thomas Hansen, Director, nodeable

Jeff Ledbetter, Product Architect, Roundtable Software



About Us - nodeable

- Roundtable experts
 - Implementations, audits, updates, customizations
 - Complementary solutions development
- Docker and DevOps experts
- Roundtable Automation
 - CI/CD and DevOps enablement

roundtable
strategic partner program

About Us - Roundtable Software

- Roundtable TSMS
 - Software Configuration Management for OpenEdge



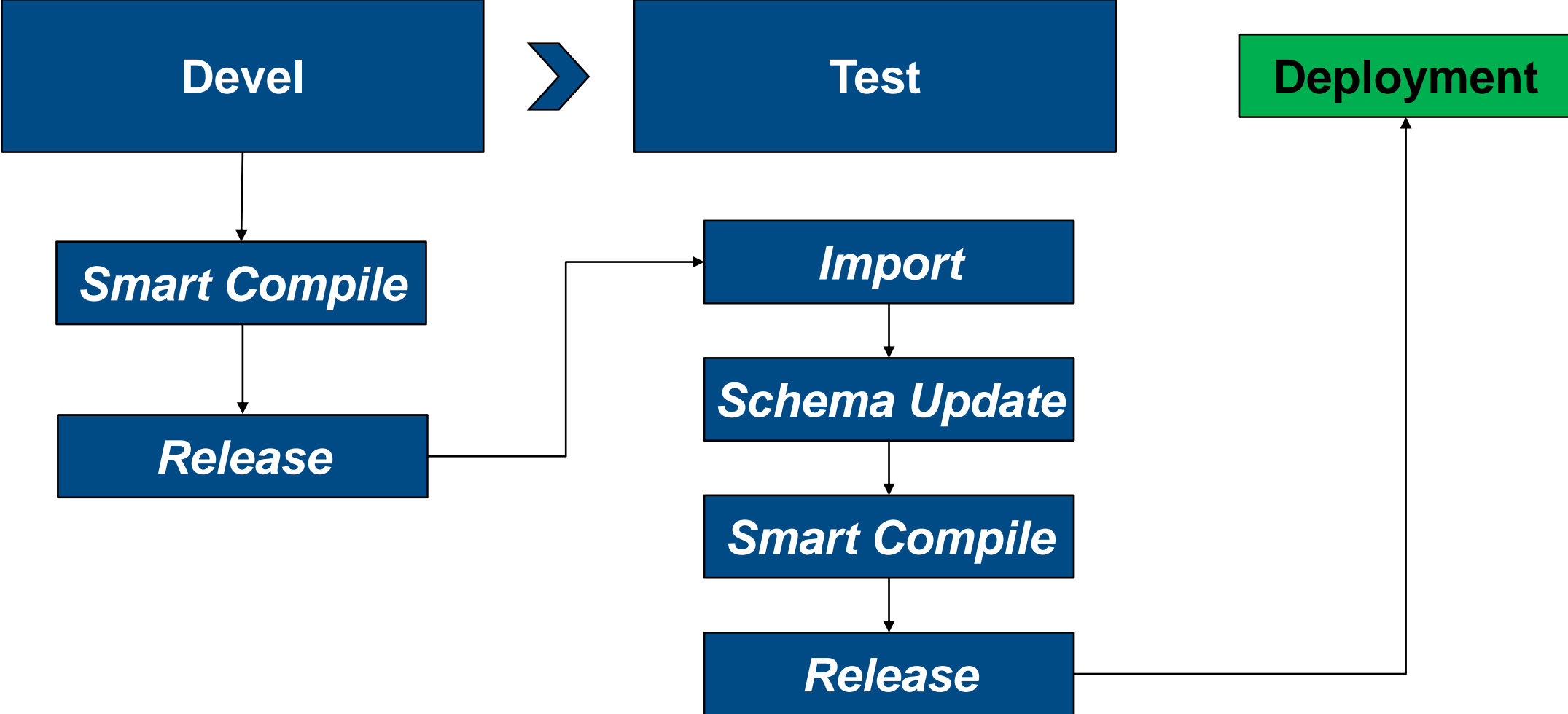
Session Agenda

- Manual Processes
- Automation Need
- The Processes
- How We did it
- The Next Step
- Review/Questions

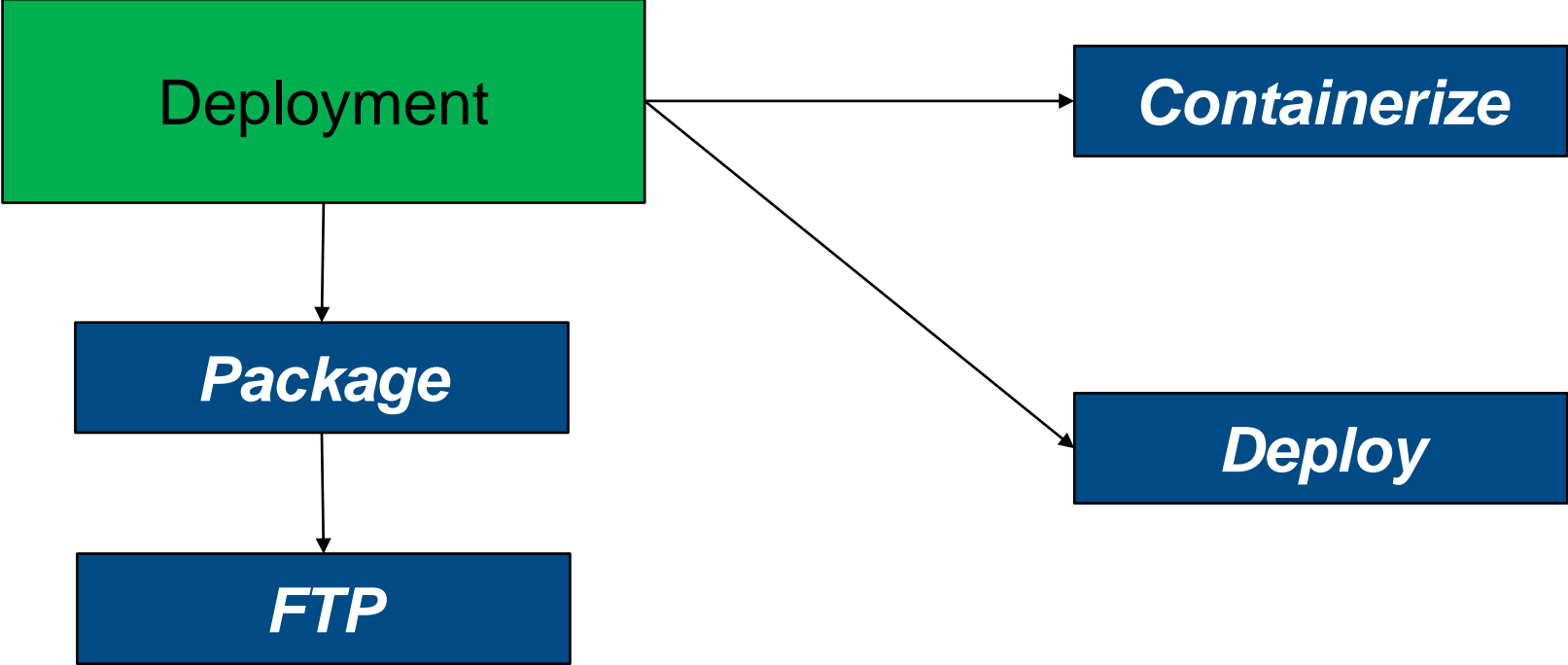
Routine / Manual Processes

- Daily builds / compilations
- Code freezes and baseline promotions
- Schema updates
- Deployments

Manual Roundtable Processes



Manual Processes – Deployment Packaging

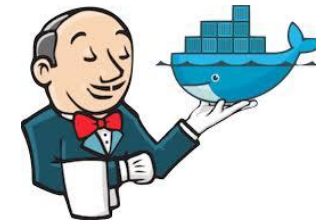


The Need for Automation

- Eliminate manual processes
- Error Reduction
- Environment integrity checking
- Continuous Integration
- Customer Updates
- Iterative Deployments



- Secure REST APIs for common Roundtable TSMS processes
 - Accessible with any HTTP(s) capable client
- Front-end Built with Docker containers
- Backend OpenEdge Automation Clients
 - Using OOABL RTB API & Proxy Wrapper



Why RTB Automation?

RTB TSMS Automated...

Faster, Easier, Safer ... Better



Why RTB Automation?

Integration with other tools & technologies

- CI/CD Tools
 - Jenkins pipelines and libraries included
 - Others can be added easily
- Issue tools
 - JIRA
- SonarQube



Why RTB Automation?

Integration with other tools & technologies

- Git integration
 - Baselines creates hybrid environment between RTB TSMS and Git repos
- Code review tools
- Build tools
 - Docker
 - External Git based CI/CD tools
- Cloud infrastructures and services
 - AWS
 - Azure



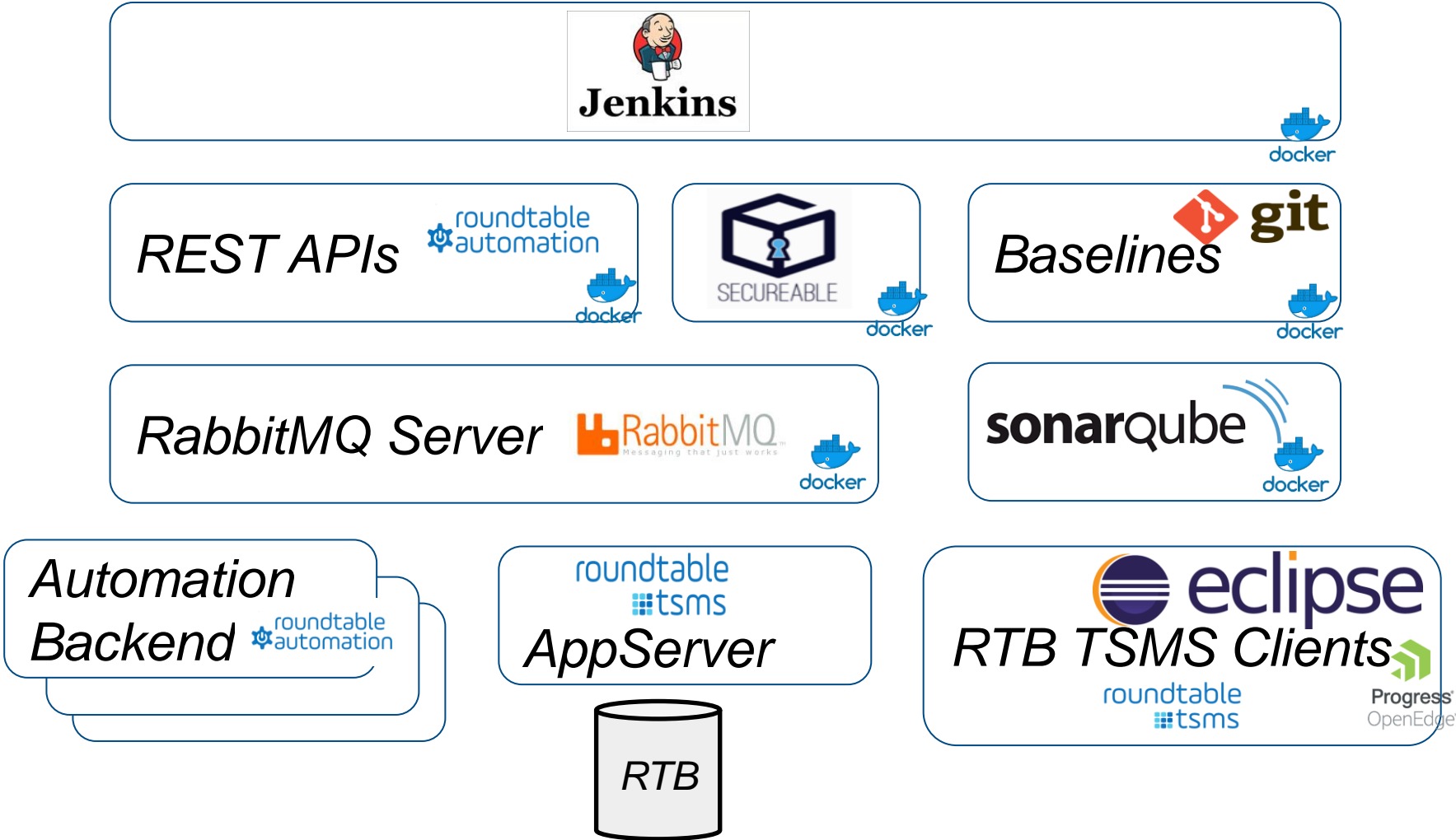
Why RTB Automation?

Enhanced auditing

- Detailed logs (not available in RTB)
- Auto-generated RTB TSMS reports



Automation Architecture



Jenkins Pipelines

- What are pipelines?
 - Groovy code for Jenkins to automate ...
 - Use standard RTB features through Automation APIs
 - Also use other tools
 - Ant & PCT
 - SonarQube
 - Install builders
 - Docker
 - ...

Jenkins Groovy

- Pipeline As Code
- Support multiple languages
 - Shell scripts
 - Windows bat / cmd
 - Windows powershell
 - Ant scripts

Groovy anyone?

```
stages {
  stage('login') {
    steps {
      retry(3) {
        withCredentials([usernamePassword(credentialsId: params.UserCredentials, passwordVariable: 'password', usernameVariable: 'userId')]) {
          script {
            token = AutomationLogin(userId, password)
          }
        }
      }
    }
  }
}
```

```
stage('create RTB release') {  
  steps {  
    // bat 'call D://Deployment/ProMark/test/Compile/SetBuildVer.cmd'  
    script {  
      props = readProperties file: 'c:/Automation/config/buildenv.props'  
      processResult = rtbAutomationRelease(  
        params.workspaces,  
        props.Beta,  
        params.createReport,  
        params.verboseOutput,  
        token)  
      // use https://wiki.jenkins.io/display/JENKINS/Groovy+Postbuild+Plugin
```



Groovy anyone?

```
withEnv(["MainVer=${props.MainVer}", "BuildVer=${props.BuildVer}", "node=${node}"]) {
    withOeEnv("${openedgeVersion}") {
        withAnt(installation: 'RTB Automation Ant') {
            script{
                antCommand = "ant -buildfile ant/compile-pro.xml -DBuildJsonProps=${params.'automation properties file'} "

                // The following options can be set in the command line and will therefore override anything in the config file. A
                if (params.'force compile' == true) {
                    antCommand = "${antCommand} -DCompileForce=true"
                }

                if (params.'delete rcode before compile' == true) {
                    antCommand = "${antCommand} -DDeleteBeforeCompile=true"
                }

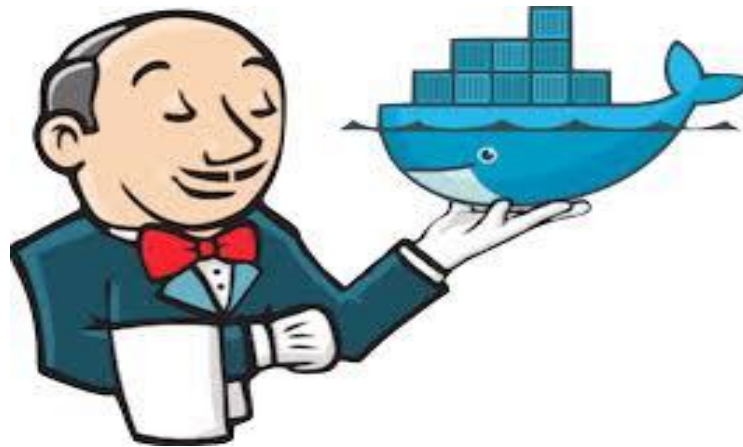
                antCommand = "${antCommand} -DCompileFailOnError=${params.'Compile fail on error'}"

                if(isUnix()) {
                    sh "${antCommand}"
                } else {
                    bat "${antCommand}"
                }
            }
        }
    }
}
```



Demo

Pipeline examples



Simple pipeline

Pipeline 1. RTB Automation Workspace Release



[Recent Changes](#)

Stage View

Average stage times:
(Average full run time: ~23s)

#3
Oct 08
17:15
No Changes

Declarative: Checkout SCM	login	create RTB release	logout
2s	6s	2s	447ms
1s	2s	2s	391ms




Complex pipeline

Pipeline BuildBeforeCodeStopPipeline

[add description](#)

Disable Project

 [Recent Changes](#)

Stage View

	Declarative: Checkout SCM	Setup	Update build version etc	JIRA Query/Update Data	RTB Changed text and layout in Devel	RTB Release in Devel	RTB Import into Test	Deployment from Test	Build Portals	Compile ProMark	Create ProMarkREST.war	Copy to Upgrade and Latest	Wise Compile	Create ISO and Zip files	Copy artifacts to Azure	Run automated tests to validate build	Declarative: Post Actions
Average stage times: (Average full run time: ~50min 39s)	3s	4s	22s	32s	1min 40s	1min 14s	2min 23s	2min 58s	5min 51s	1min 27s	24s	10s	3min 0s	1min 3s	10min 14s	482ms	7s
#464 Oct 08 10:23 3 commits	3s									1min 33s	1min 11s	31s	8min 11s	2min 26s	16min 5s	1s	6s
#463 Oct 08 09:57 3 commits	6s	3s	22s	35s	1min 42s	1min 17s	2min 29s	2min 55s	5min 29s	1min 30s failed	69ms failed	58ms failed	58ms failed	61ms failed	60ms failed	55ms failed	6s
#462 Oct 08 03:24 No Changes	3s	4s	21s	26s	1min 39s	1min 8s	2min 20s	3min 1s	5min 39s	1min 58s	1min 5s	26s	8min 9s	3min 45s	21min 37s	252ms	6s



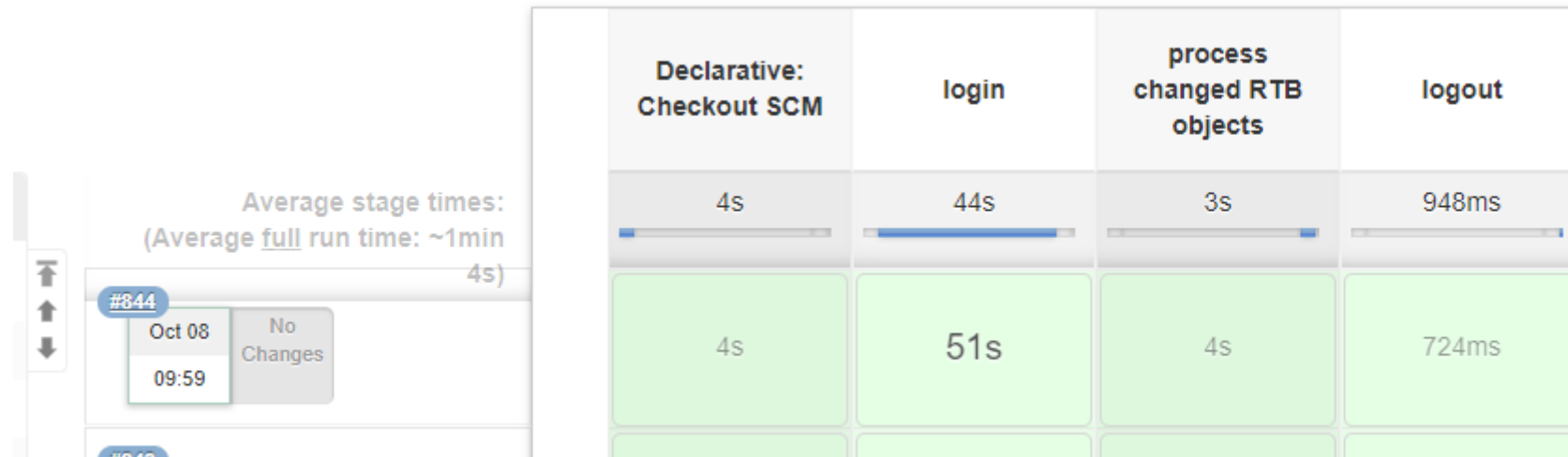
Custom RTB pipeline

Pipeline 1 - RTB Automation Changed text and layout



[Recent Changes](#)

Stage View



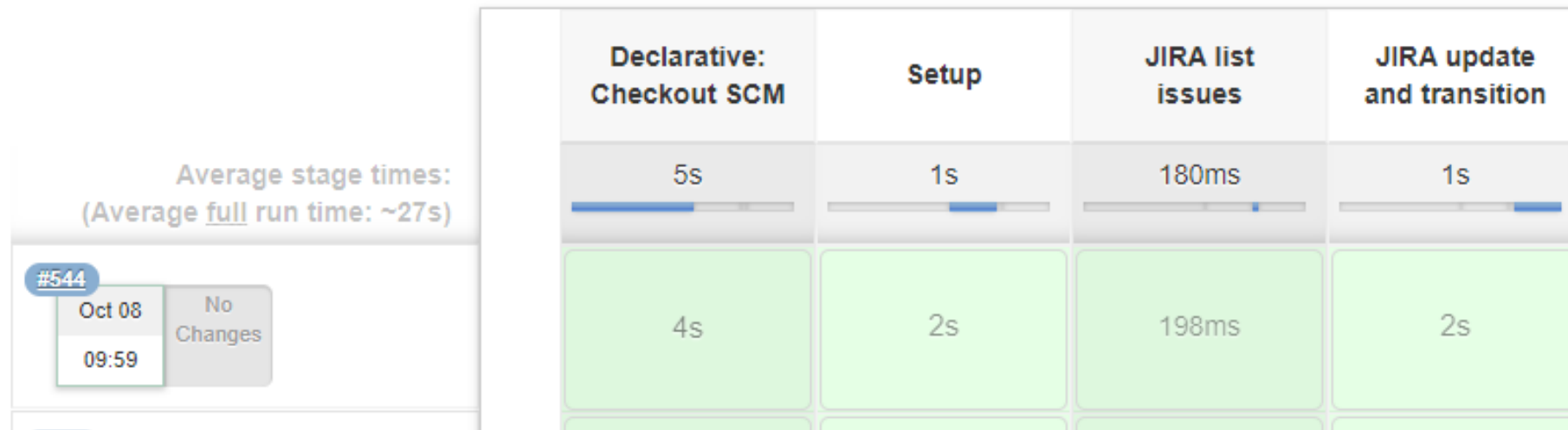
JIRA pipeline

Pipeline JIRA List and Update Issues














[Recent Changes](#)

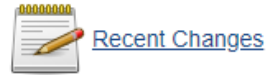
Stage View



PCT Compile and SonarQube Scan pipeline

-  [Back to Dashboard](#)
-  [Status](#)
-  [Changes](#)
-  [Build with Parameters](#)
-  [Delete Pipeline](#)
-  [Configure](#)
-  [Full Stage View](#)
-  [SonarQube](#)
-  [Open Blue Ocean](#)
-  [Rename](#)
-  [Pipeline Syntax](#)

Pipeline CompileAndSonarScanPipeline



Stage View

Average stage times:
(Average full run time: ~12min
47s)

Build #	Date	Commits
#61	Oct 08 10:54	3 commits
#60	Oct 08 04:17	2 commits
#59		

	Declarative: Checkout SCM	Git checkouts	PCT Build: Compile PRO	SonarQube analysis	Declarative: Post Actions
	3s	21s	5min 0s	3min 45s	7s
#61	3s	29s	7min 4s	5min 2s	
#60	2s	26s	6min 53s	5min 18s	
#59					



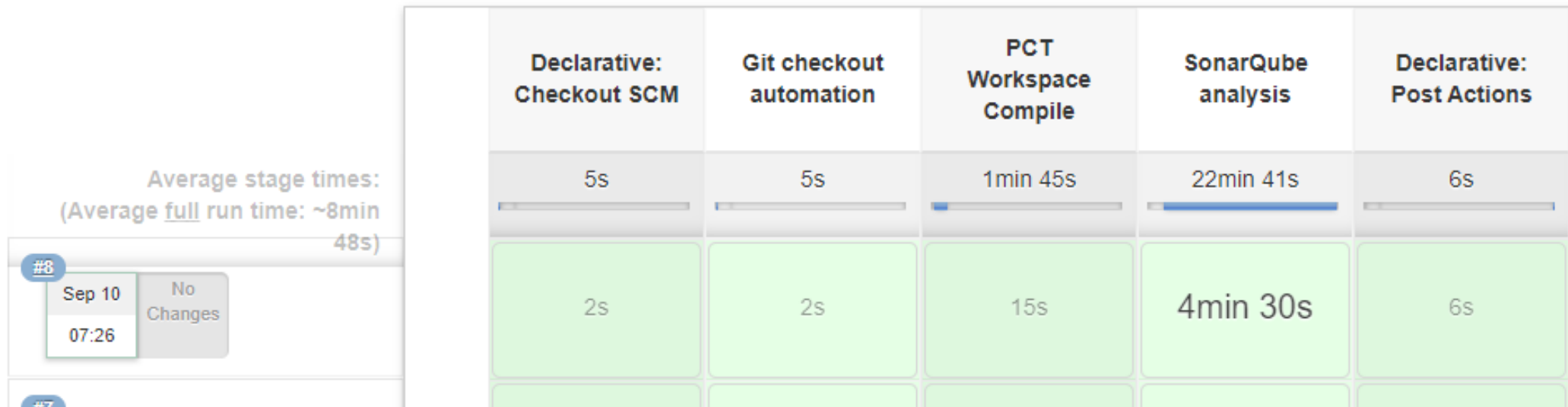
Workspace PCT Compile and SonarQube Scan pipeline

Pipeline WorkspacePCTCompileAndSonarScanPipeline



[Recent Changes](#)

Stage View



PASOE Web Application pipeline

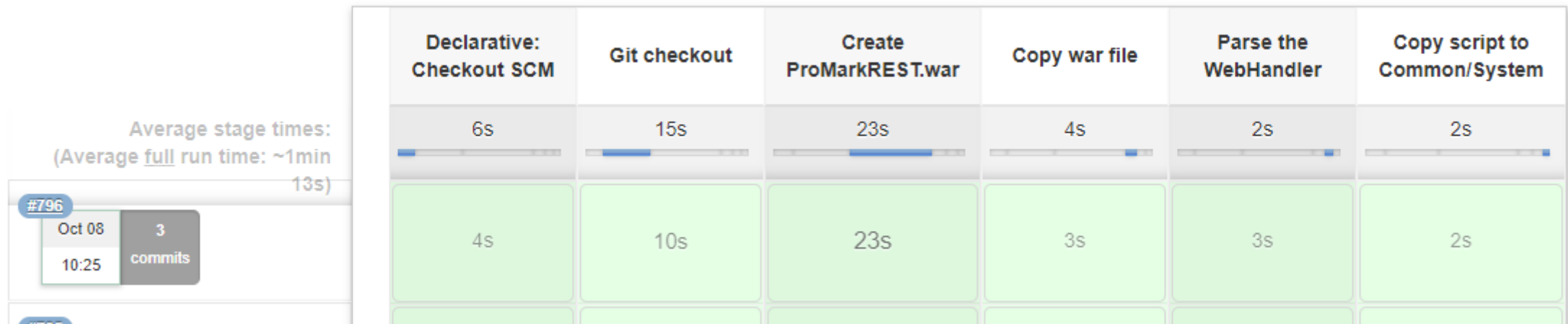
Pipeline CreateProMarkRestWarPipeline

Create the .war file based on the code in Bitbucket



[Recent Changes](#)

Stage View



Custom Automation Processes

- Custom RTB Processes
 - Standard API for custom processes
 - Backend process and properties (json) can be written in ABL
 - Using standard RTB APIs
 - RTB Automation “ProxyHelper” classes included




Baselines

- “Magic deployments”
- Link RTB Workspaces to Remote Git repositories
- “Build” creates special RTB deployments packaged and converted into Git branch commits
 - Git branches can be used for
 - Code Reviews
 - CI/CD builds
 - SonarQube code scans
 - Container builds
 - Further processing using Git-based branching and workflows



Baselines

 **nodeable** / dashboard-dev










Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 2](#) [Pull Requests 1](#) [Releases 0](#) [Wiki](#) [Activity](#) [Settings](#)

No Description
[Manage Topics](#)

28 Commits 6 Branches

[Branch: dashboard-dev](#) dashboard-dev [New File](#) [Upload File](#) [HTTPS](#) [SSH](#) <https://gitea.nodeable.io/nor> [Copy](#) [Download](#)


 thomas ea050908f6 Fix baselines. Auto release : 02/11/2019 22:54:02+01:00 7 months ago
 ant Automation updates 1 year ago
 appSolutions/appSolute/RTB Build after updates to temp-dir stuff 8 months ago
 bin initial commit 1 year ago
 config test again 8 months ago
 curl initial commit 1 year ago
 dotr test again 8 months ago
 nodeable Build after updates to temp-dir stuff 8 months ago
 rtb_dbup update 1 year ago



Baselines

Fix baselines. Auto release : 02/11/2019 22:54:02+01:00

[Browse Source](#)

 thomas 7 months ago

parent [995bc78dae](#) commit [ea050908f6](#)

1 changed files with 1 additions and 1 deletions

[Unified View](#) [Show Diff Stats](#)

+1 -1 scm/rtb/engine/proxyapi/controllers/base/baselines.cls [View File](#)

604	IF oDeployment:has("releaseNote":U) THEN	604	IF oDeployment:has("releaseNote":U) THEN
605	DO:	605	DO:
606	ASSIGN	606	ASSIGN
607	cCommitMessage = cCommitMessage + "~n":U + oDeployment:GetCharacter("releaseNote":U).	607	cCommitMessage = TRIM(cCommitMessage + "~n":U + oDeployment:GetCharacter("releaseNote":U)).
608	END.	608	END.
609		609	
610	LogHelper:WriteMessage(SUBSTITUTE(" Uploading file '&1'...", cUploadFile)).	610	LogHelper:WriteMessage(SUBSTITUTE(" Uploading file '&1'...", cUploadFile)).



PUGCHALLENGE

AMERICAS

Questions?

Thomas Hansen – thomas@nodeable.io

Jeff Ledbetter – jeff.ledbetter@roundtable-software.com

nodeable

