

测试工具开发中的持续集成与持续交付实践

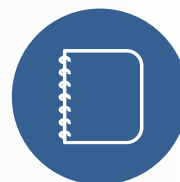
周运杰 2017-07-19

下研发
软件研发
SOFTWARE
DEVELOPMENT

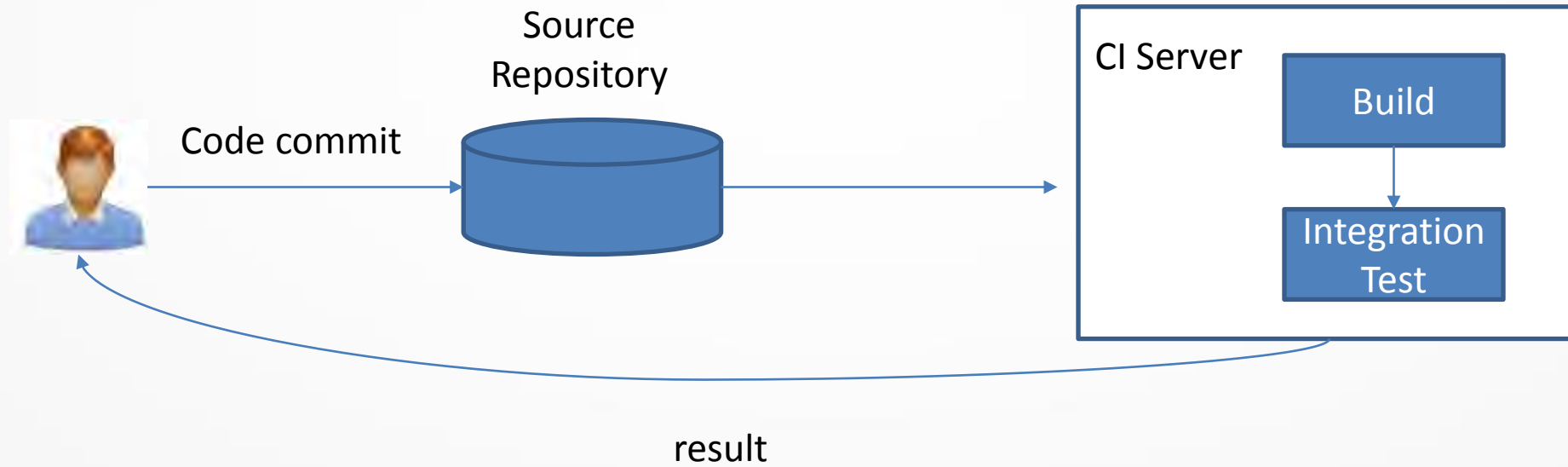
一、什么是CI/CD/DevOps

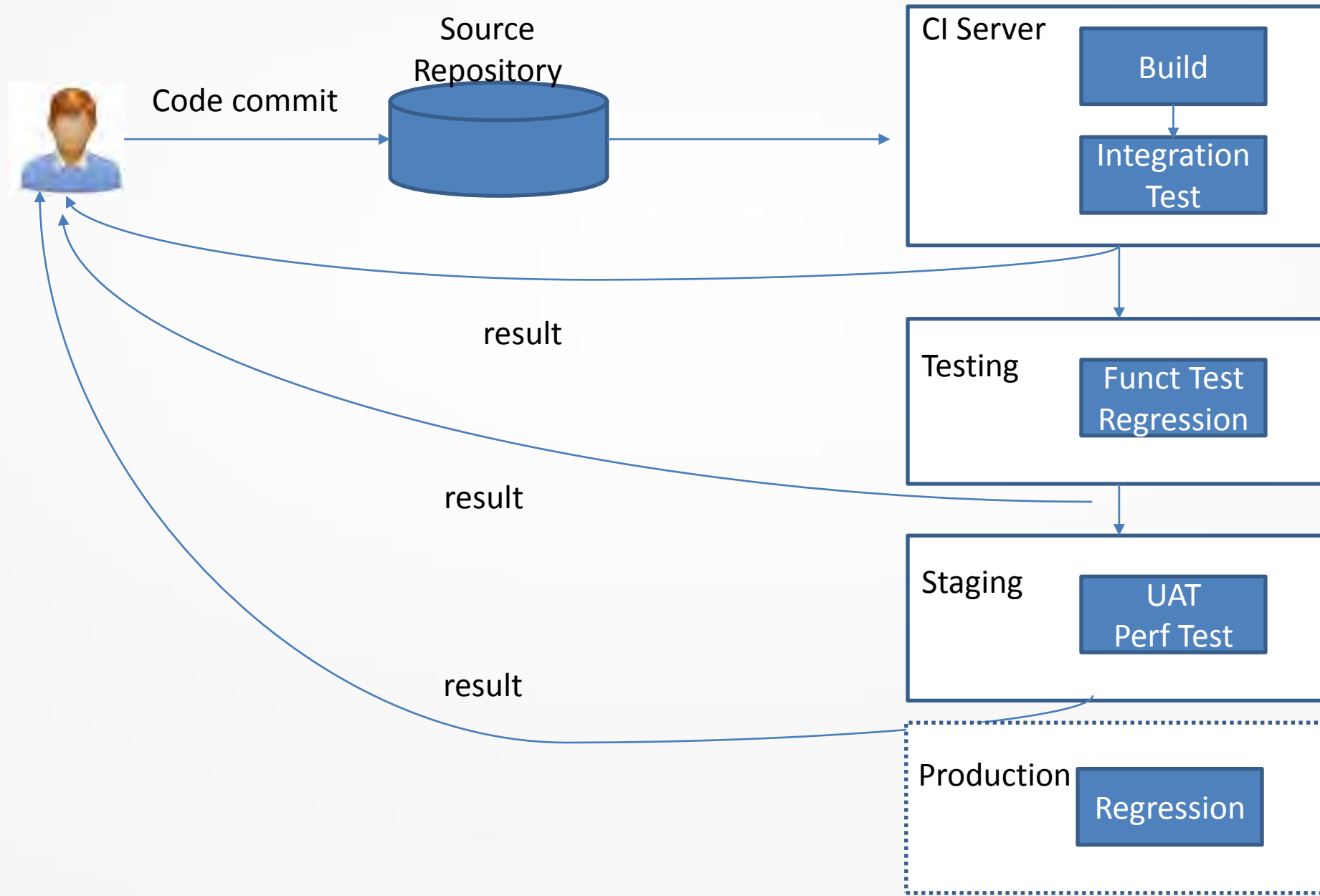
二、Jenkins Pipeline及端到端交付

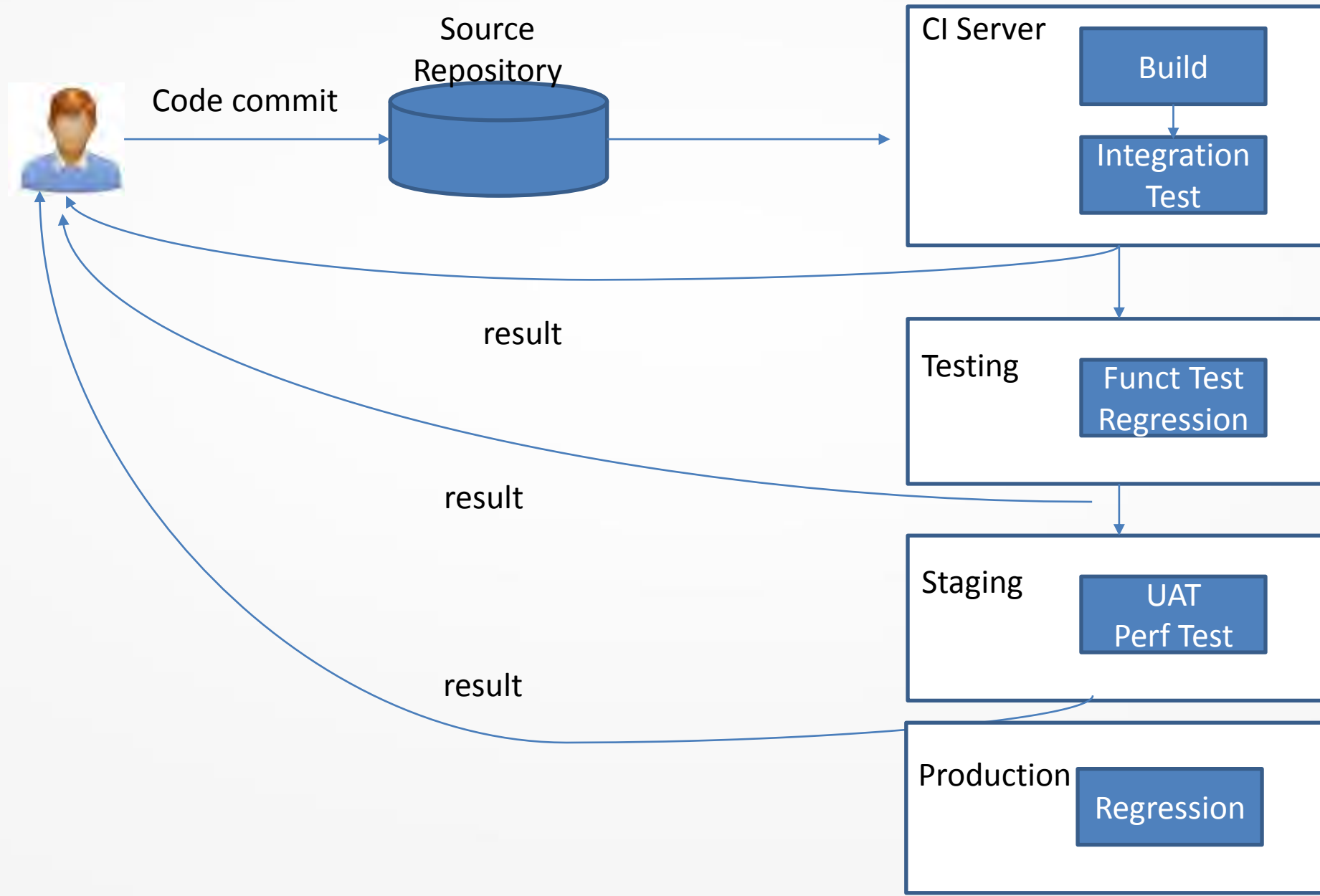
三、测试工具开发中CI/CD的落地



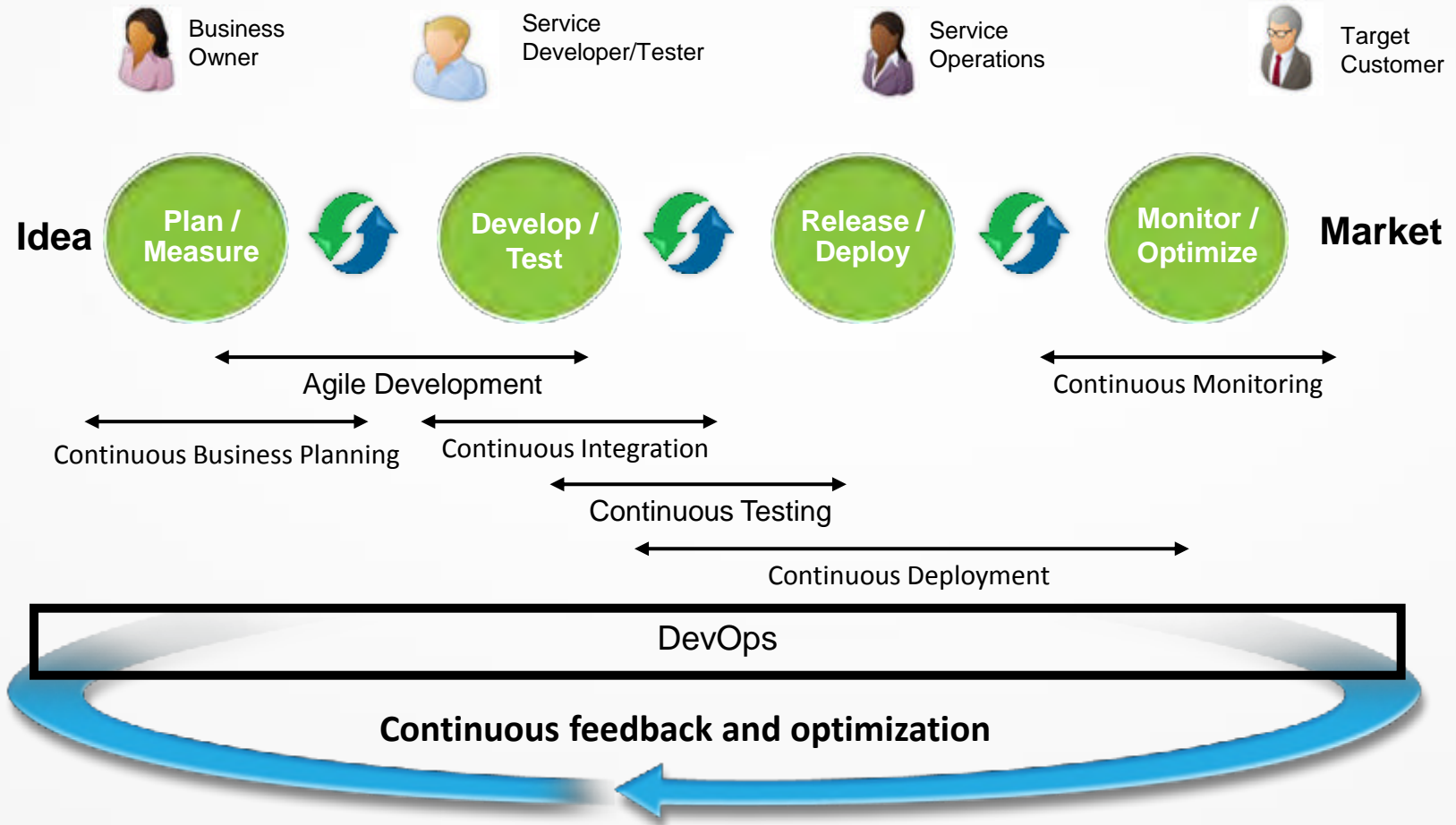
一、什么是CI/CD/DevOps



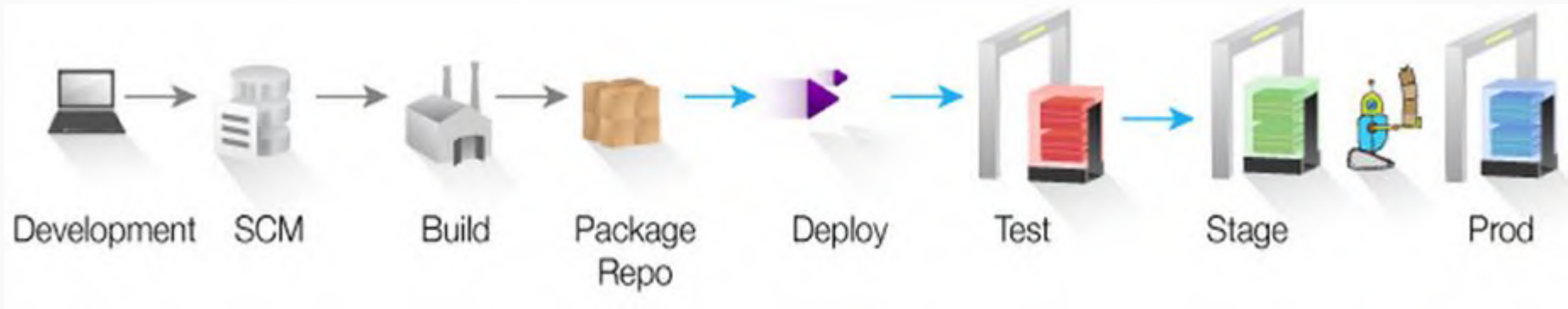




DevOps 生命周期



DevOps Toolset



maven

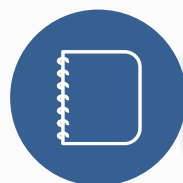
JUnit



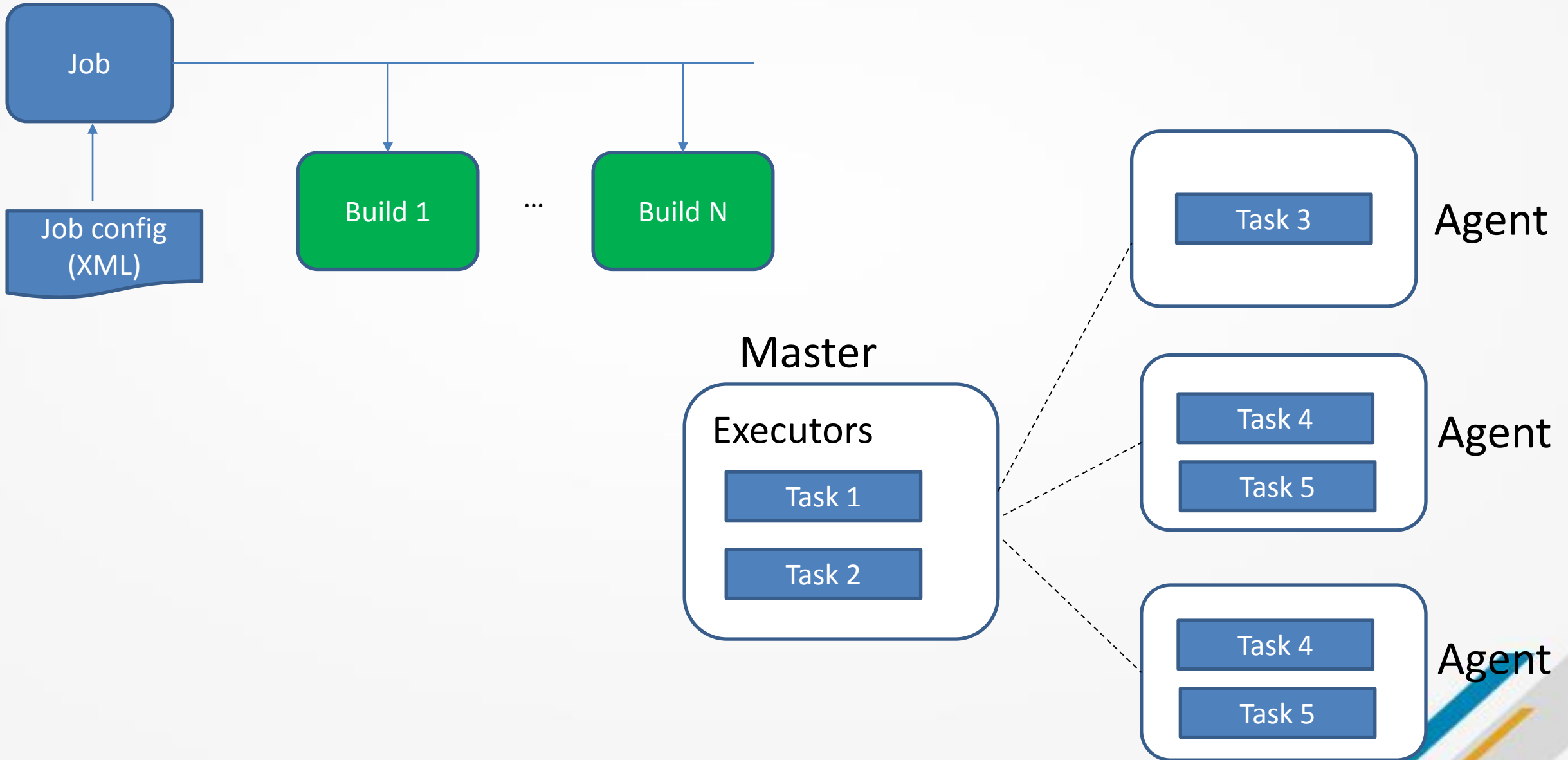
Nexus

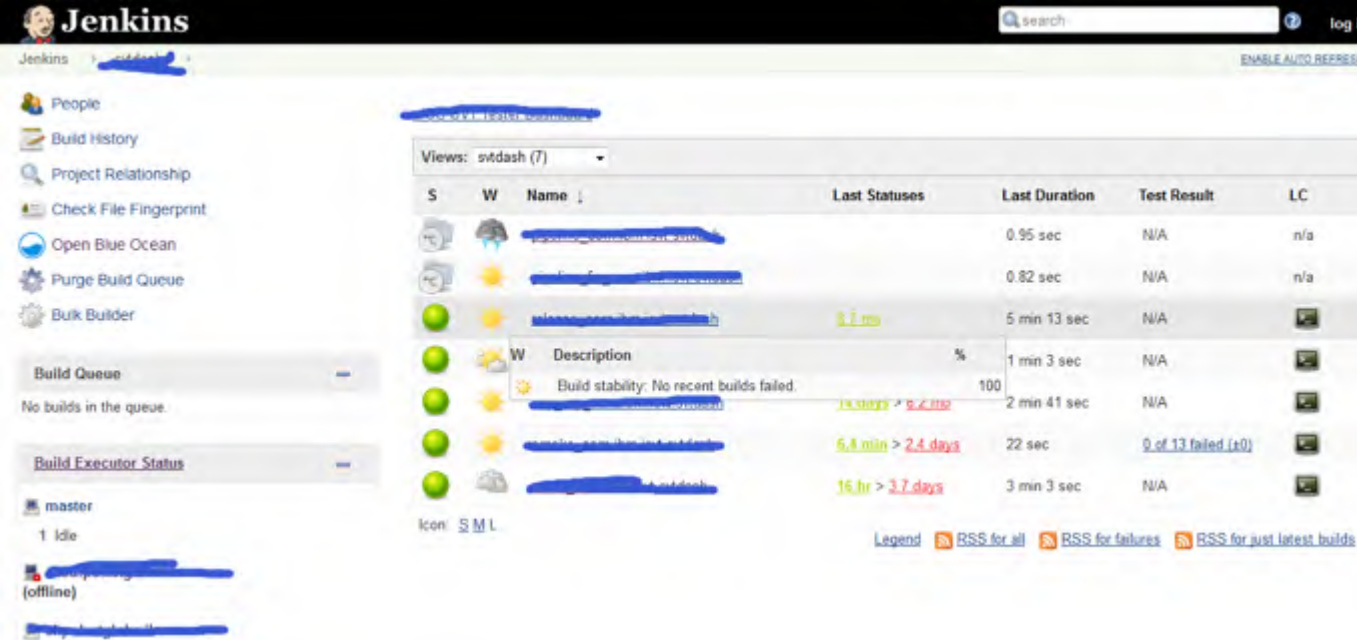


Nagios



二、 Jenkins Pipeline及端到端交互





The screenshot shows the Jenkins web interface. On the left, there is a sidebar with navigation options like 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Open Blue Ocean', 'Purge Build Queue', 'Bulk Builder', 'Build Queue', and 'Build Executor Status'. The main area displays a table of build history for a project named 'svtdash (7)'. The table has columns for 'S' (status), 'W' (workspace), 'Name', 'Last Statuses', 'Last Duration', 'Test Result', and 'LC' (last commit). The table shows several builds, with the most recent one having a duration of 5 min 13 sec and a test result of 'N/A'. Below the table, there is a legend for RSS feeds and a link to 'S.M.L.'.

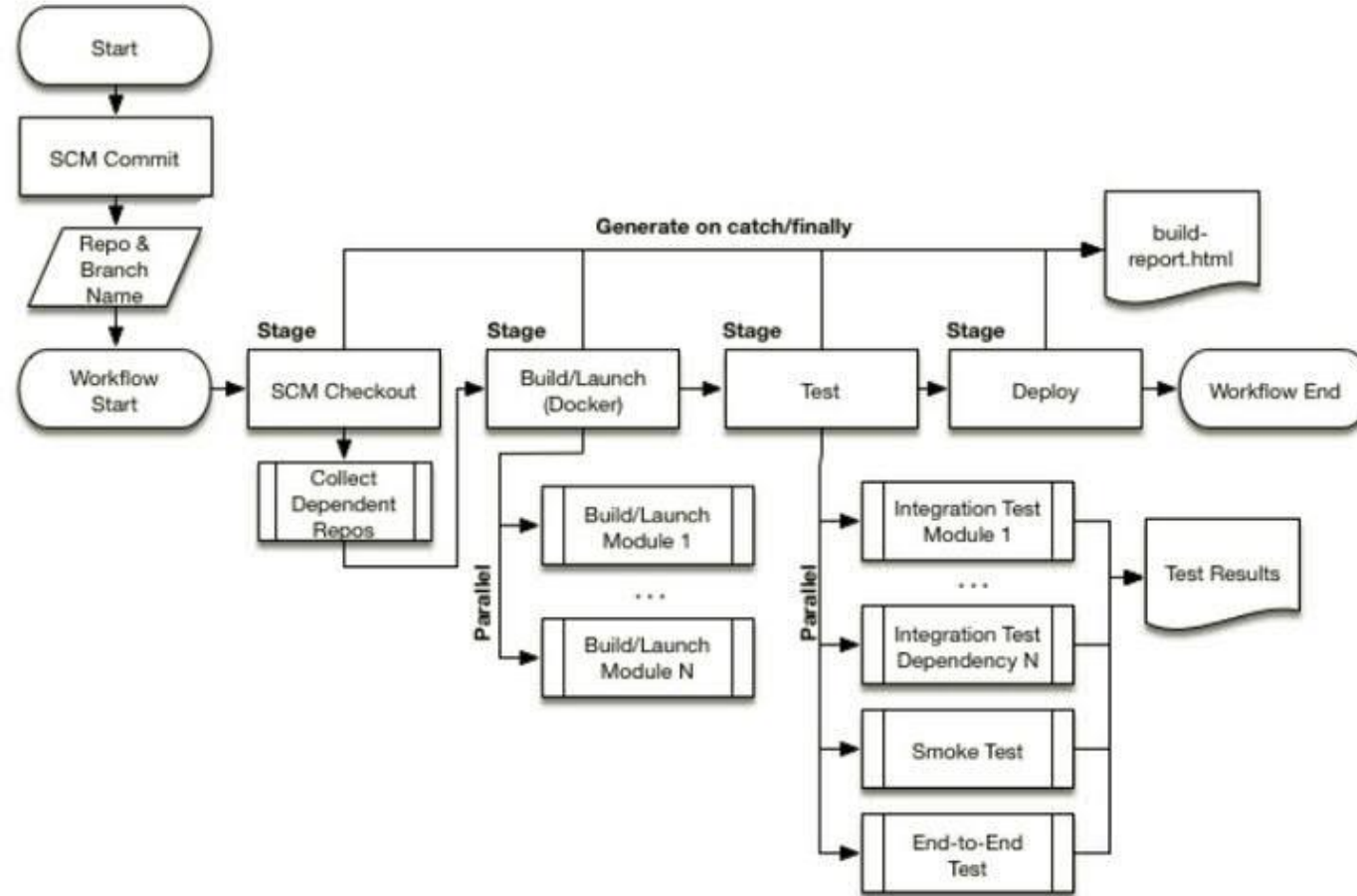
S	W	Name	Last Statuses	Last Duration	Test Result	LC
🟢	🌤️	[redacted]		0.95 sec	N/A	n/a
🟢	🌤️	[redacted]		0.82 sec	N/A	n/a
🟢	🌤️	[redacted]		5 min 13 sec	N/A	[redacted]
🟢	🌤️	W Description	%	1 min 3 sec	N/A	[redacted]
🟢	🌤️	Build stability: No recent builds failed.	100	2 min 41 sec	N/A	[redacted]
🟢	🌤️	[redacted]		22 sec	0 of 13 failed (x0)	[redacted]
🟢	🌤️	[redacted]		3 min 3 sec	N/A	[redacted]

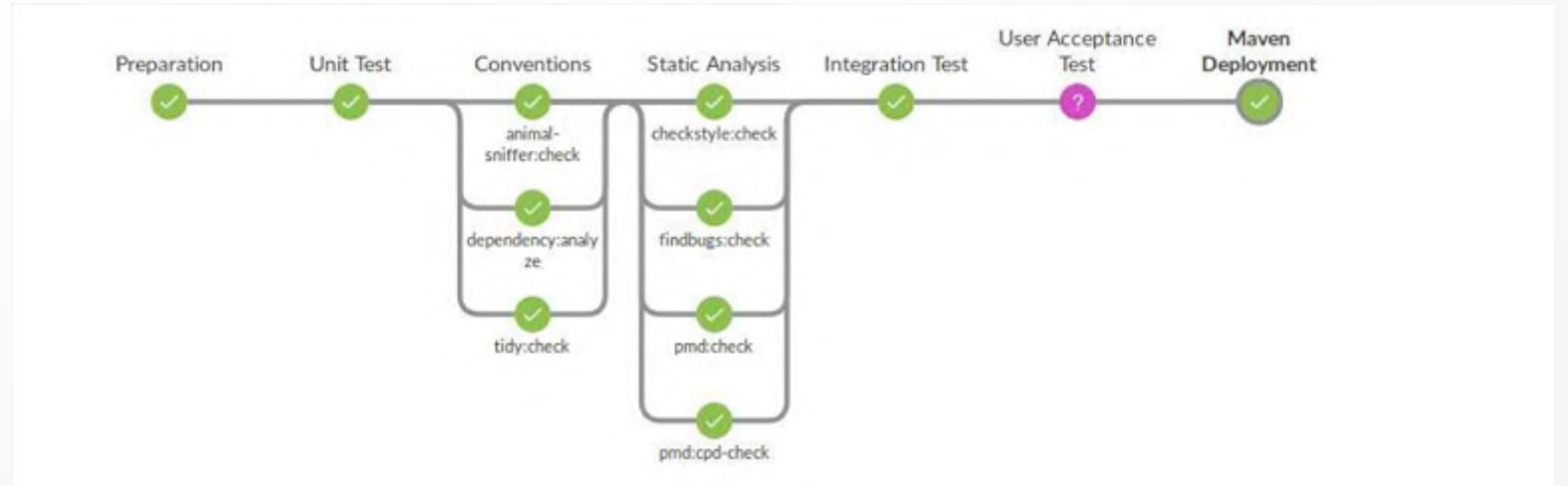
- Build
- Static Checking
- Release
- Deploy
- UAT

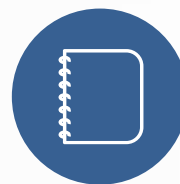
Jenkinsfile

```
#!/groovy

pipeline {
  stages {
    stage ('Preparation') {
      steps {
        ...
      }
    }
    stage ('Unit Test') {
      steps {
        ...
      }
    }
    stage ('Conventions') {
      ...
    }
    stage ('Static Analysis') {
      ...
    }
    stage ('Integration Test') {
      ...
    }
    stage ('User Acceptance Test') {
      ...
    }
  }
}
```







三、测试工具开发中CI/CD的落地

敏捷化的项目管理

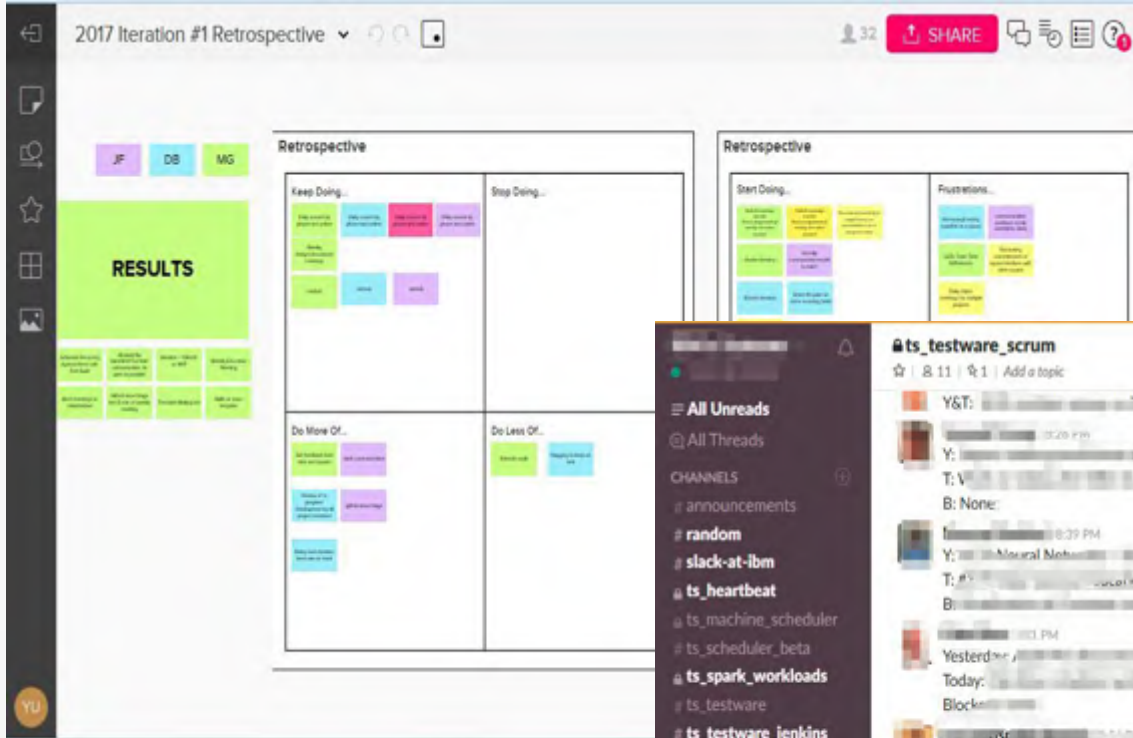
代码的持续集成

代码的静态扫描

持续测试

环境配置

持续监控



2017 Iteration #1 Retrospective

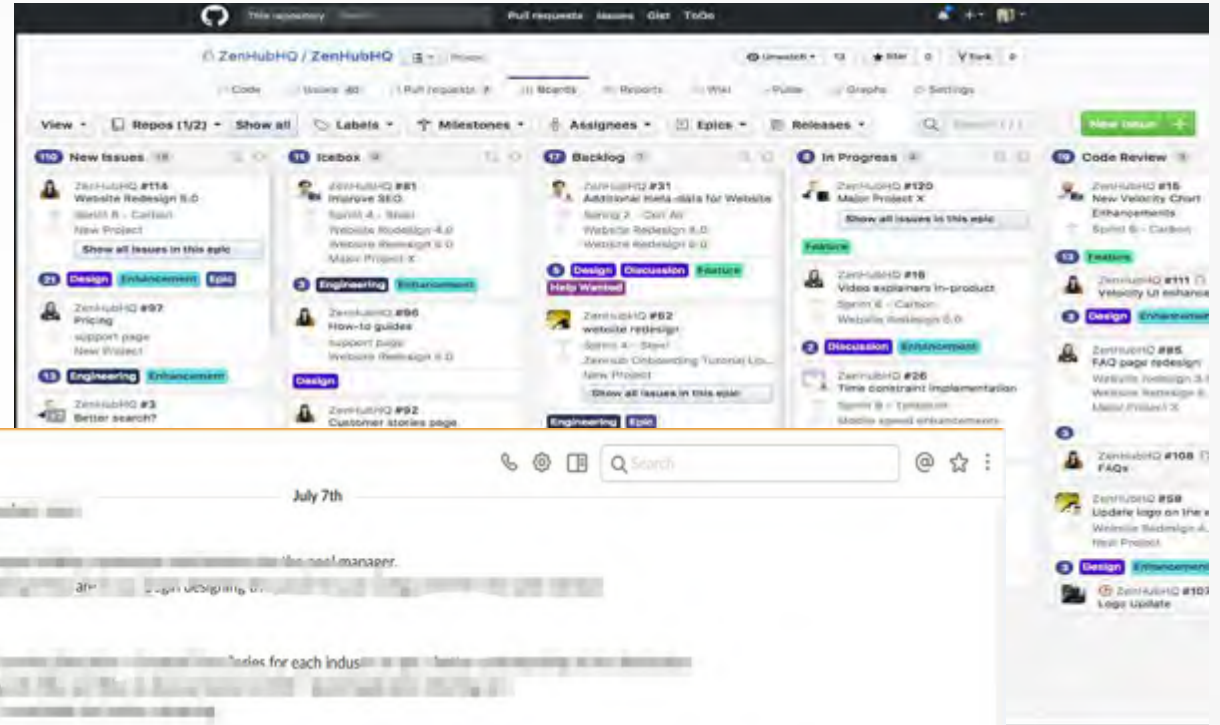
RESULTS

Retrospective

Keep Doing...	Stop Doing...
Start Doing...	Frustrations...

Do More Of...

Do Less Of...

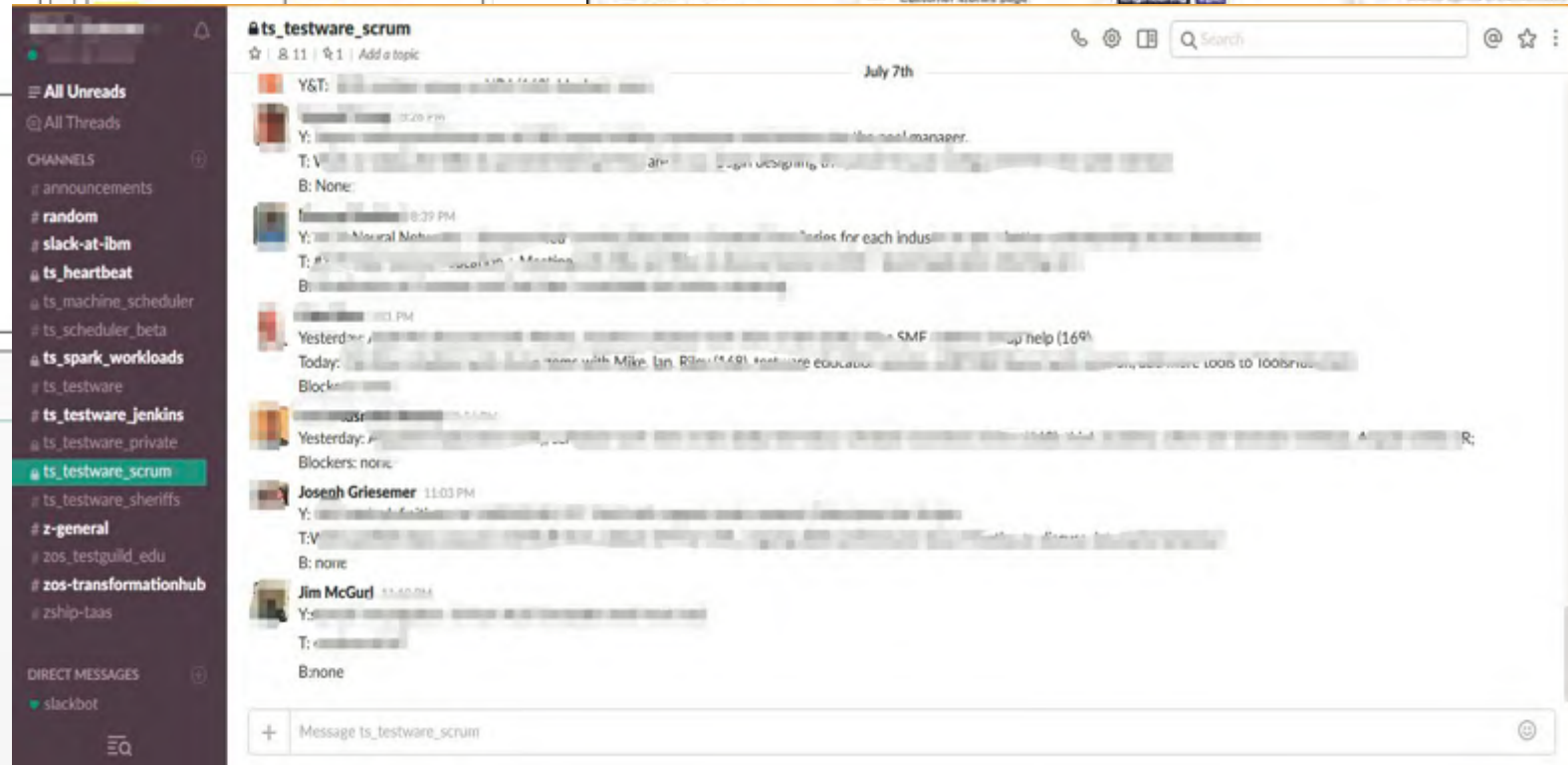


ZenHubHQ / ZenHubHQ

View - Repos (1/2) - Show all

Labels - Milestones - Assignees - Epics - Releases

- 10 New Issues**
- 11 Icebox**
- 17 Backlog**
- 4 In Progress**
- 10 Code Review**



#ts_testware_scrum

July 7th

Y&T: ...

Y: ...

T: V: ...

B: None

Y: ...

T: ...

B: ...

Yesterday: ...

Today: ...

Blockers: ...

Yesterday: ...

Blockers: none

Joseph Griesemer 11:03 PM

Y: ...

T: V: ...

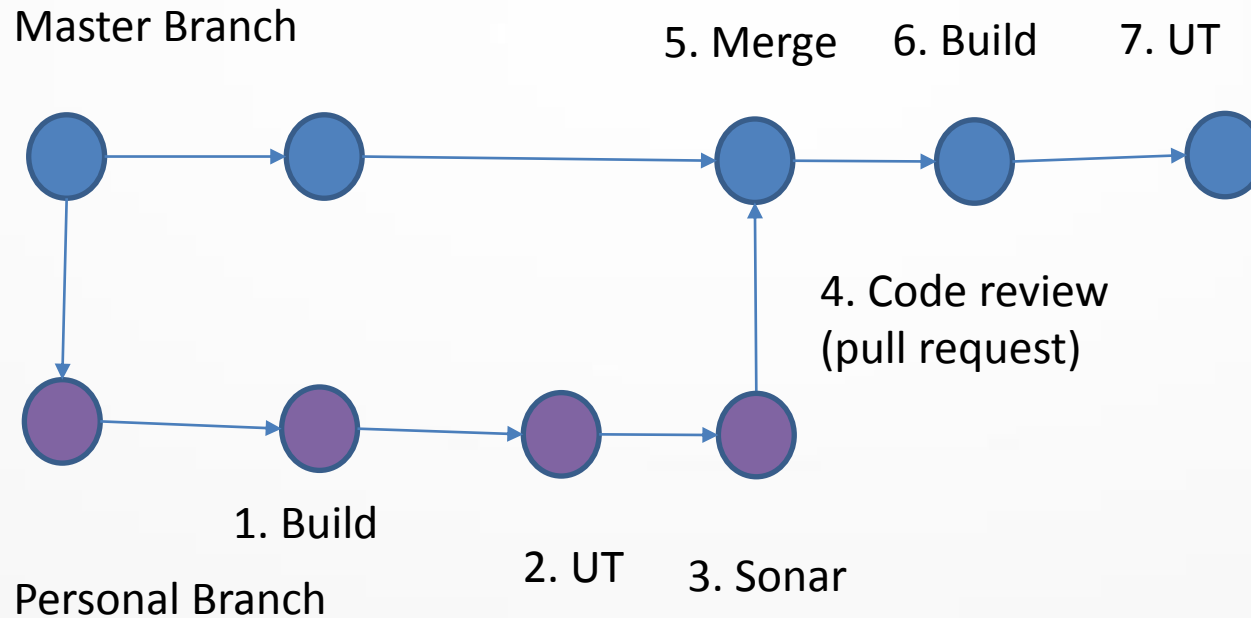
B: none

Jim McGur 11:40 PM

Y: ...

T: ...

B: none



[Code](#) | [Pull requests 0](#) | [Projects 0](#) | [Boards](#) | [Burndown](#) | [Pulse](#) | [Graphs](#) | [Settings](#)

Options

Collaborators & teams

Branches

Hooks & services

Deploy keys

Custom tabs

Services / Manage Jenkins (GitHub plugin) Test service

Jenkins is a popular continuous integration server.

Using the Jenkins GitHub Plugin you can automatically trigger build jobs when pushes are made to GitHub.

Install Notes

1. "Jenkins Hook URL" is the URL of your Jenkins server's webhook endpoint. For example: <http://ci.jenkins-ci.org/github-webhook/>.

For more information see <https://wiki.jenkins-ci.org/display/JENKINS/GitHub+plugin>.

Jenkins hook url

Active
We will run this service when an event is triggered.

Update service Delete service

Add source ▾

Build Configuration

Mode

Script Path

Scan Multibranch Pipeline Triggers

Periodically if not otherwise run

Interval

Orphaned Item Strategy

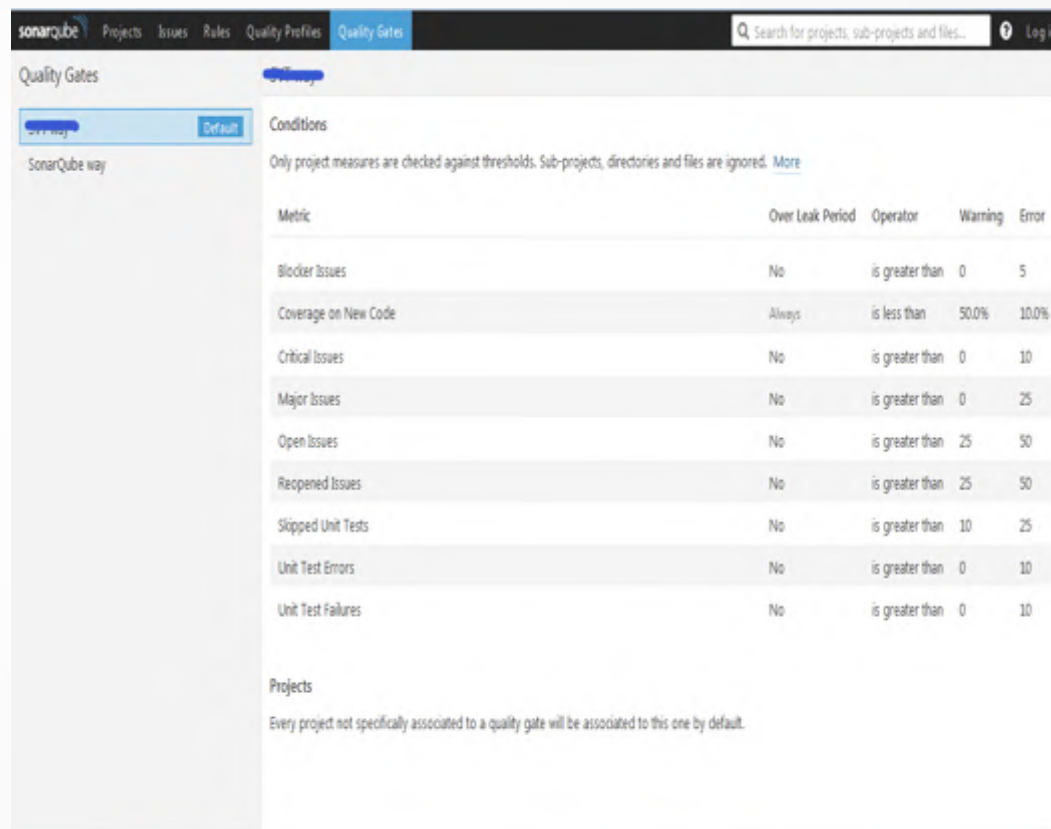
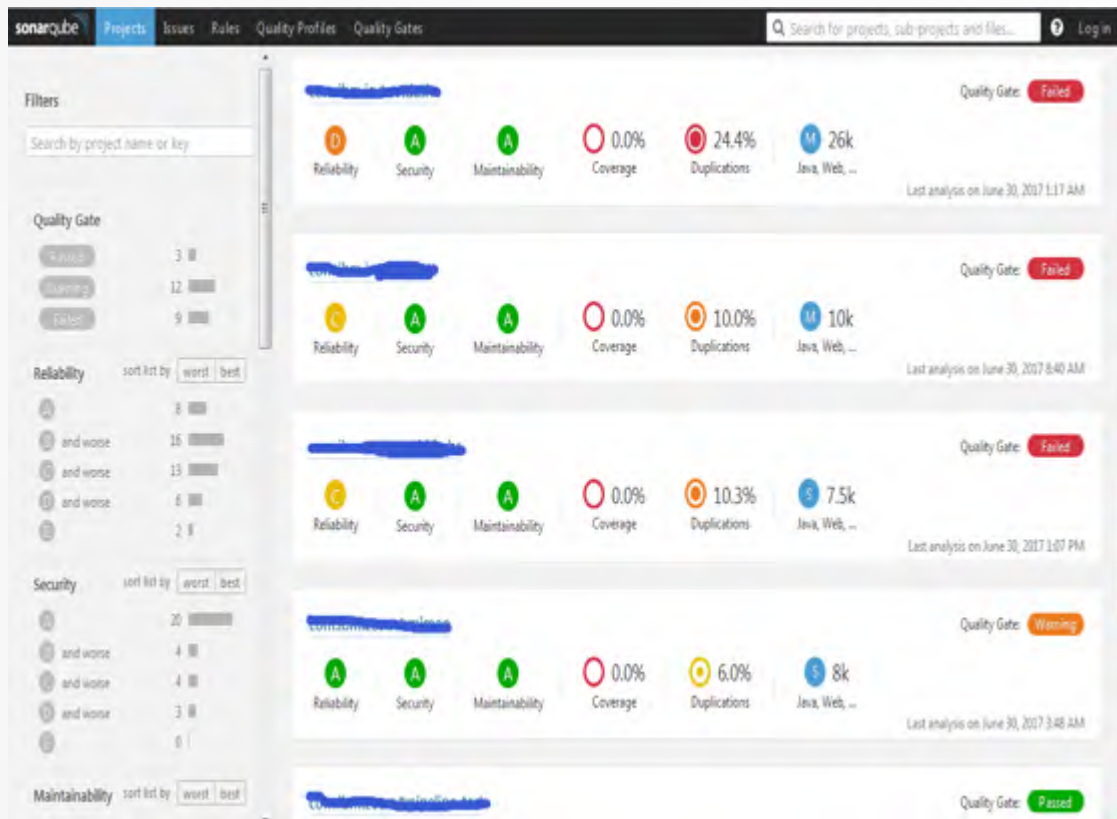
Discard old items

Days to keep old items

if not empty, old items are only kept up to this number of days

Max # of old items to keep

if not empty, only up to this number of old items are kept



集成方式:

1. Maven 插件
2. Jenkins 插件

```
stages {

  stage ('Preparation') {
    steps {
      checkout scm
      sh "mvn $MVN_OPTS clean"
    }
  }

  stage ('Unit Test') {
    steps {
      sh "mvn $MVN_OPTS install"
    }

    post {
      always {
        archive '**/target/*.jar'
        archive '**/target/*.war'
        junit '**/target/surefire-reports/*.xml'
      }
    }
  }
}
```

```
stage ('Static Analysis') {
  when {
    expression {
      return !params.SKIP_STATIC_ANALYSIS
    }
  }

  steps {
    parallel(
      'checkstyle:check': {
        sh "mvn $MVN_OPTS -Dcheckstyle.failOnViolation=false -Dcheckstyle.failOnError=false checkstyle:check"
      },
      'findbugs:check': {
        sh "mvn $MVN_OPTS -Dfindbugs.failOnError=false findbugs:check"
      },
      'pmd:check': {
        sh "mvn $MVN_OPTS -Dpmd.failOnViolation=false pmd:check"
      },
      'pmd:cpd-check': {
        sh "mvn $MVN_OPTS -Dcpd.failOnViolation=false pmd:cpd-check"
      }
    )
  }
}
```

测试目标： 模块与模块之间的接口

Java EE: 依赖注入、事务控制、访问数据库

两种方式:

Embedded

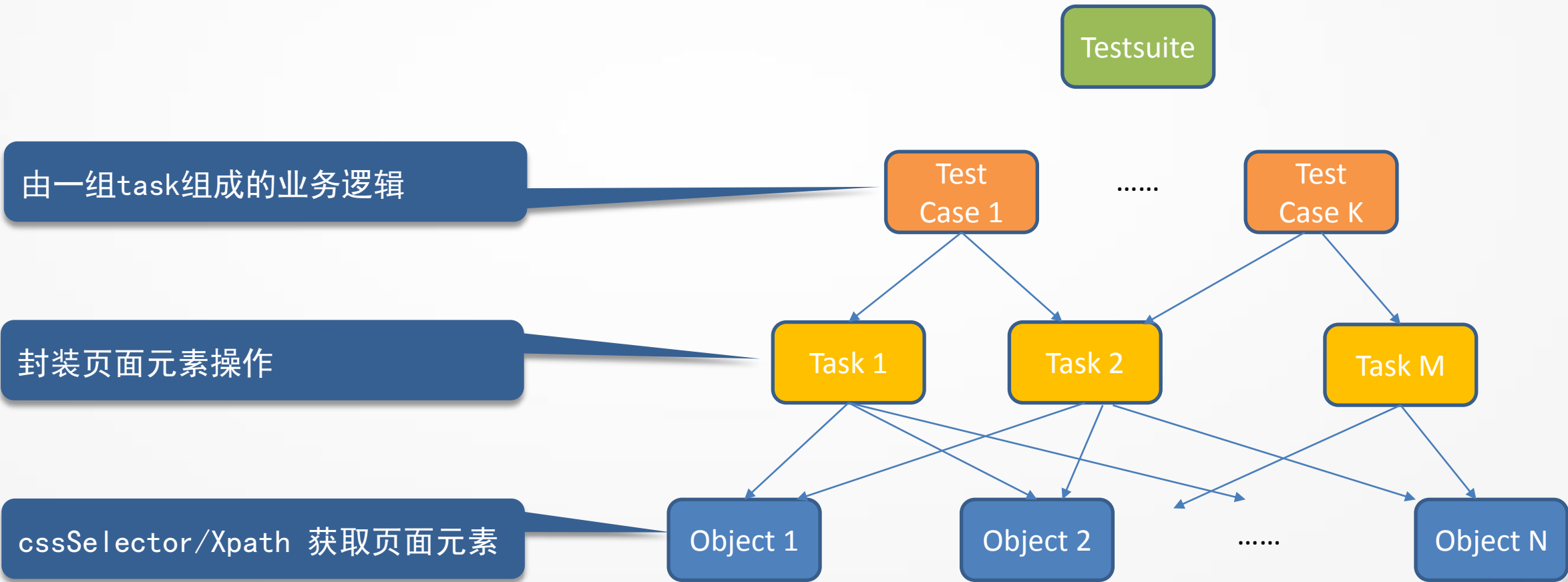
– Embedded 应用服务器 + Embedded DB

独立测试服务器



The Arquillian test infrastructure





```
stage ('Integration Test') {
  when {
    expression {
      return !params.SKIP_INTEGRATION_TEST
    }
  }

  steps {
    sh "mvn $MVN_OPTS --activate-profiles run-it"
  }

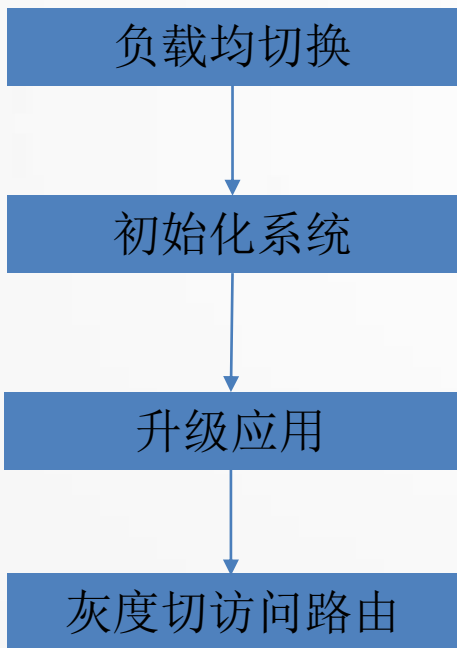
  post {
    always {
      junit '**/target/surefire-reports/*.xml'
    }
  }
}
```

```
stage ('User Acceptance Test') {
  when {
    expression {
      return !params.SKIP_USER_ACCEPTANCE_TEST
    }
  }

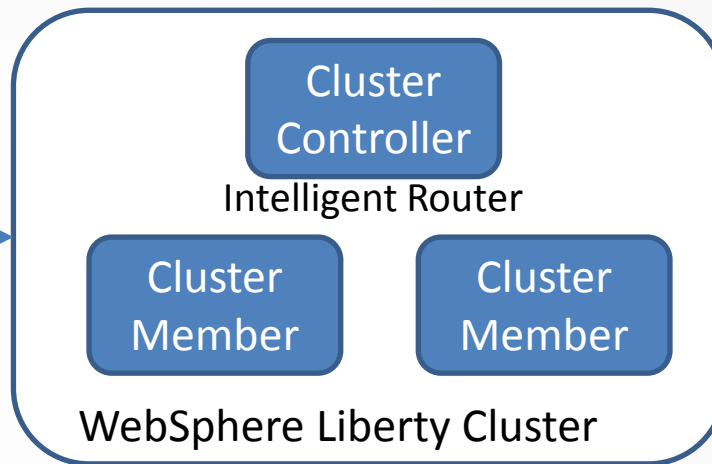
  steps {
    sh "mvn $MVN_OPTS --activate-profiles run-uat"
  }

  post {
    always {
      junit '**/target/surefire-reports/*.xml'
    }
  }
}
```


部署阶段



SCP War



```

stage 'Promotion' {
  timeout(time: 1, unit: 'HOURS') {
    input 'Deploy to Production?'
  }
}

stage 'Deploy to Production' {
  sh 'scp *.war dashboard@{SERVER}:{HOME}/uploads'
}
    
```




- Stop cluster member A
- Deploy app
- Start cluster member A

- Stop cluster member B
- Deploy app
- Start cluster member B

Pipeline 执行结果

Pipeline master

Full project name: pipeline_com.ibm.isvt.svtdash/master

 [Last Successful Artifacts](#)
[Expand all](#) [Collapse all](#)
[+ View](#)

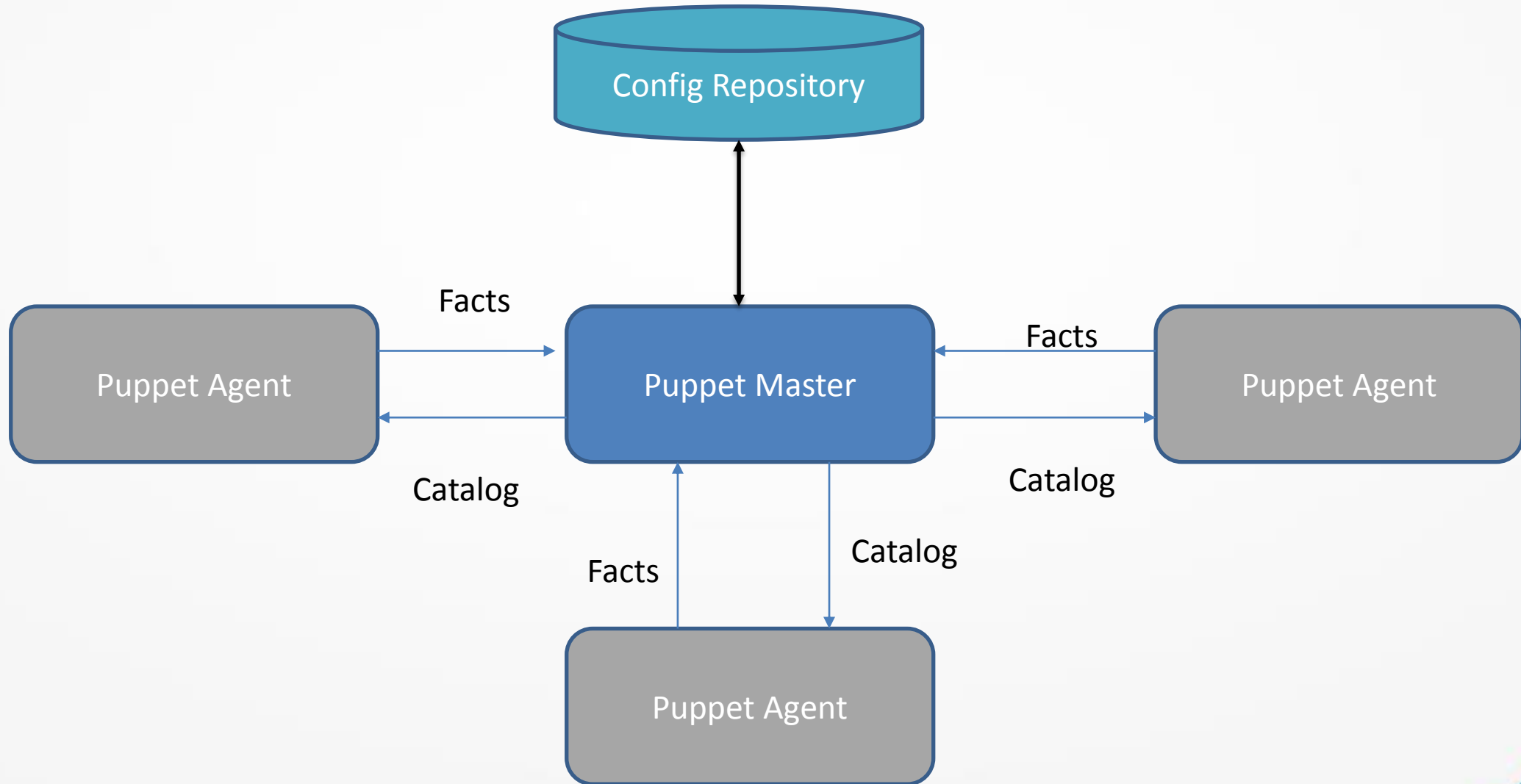
 [Recent Changes](#)



Stage View

Average stage times:

	Declarative: Checkout SCM	Declarative: Tool Install	Preparation	Unit Test	Conventions	Static Analysis	Integration Test	User Acceptance Test	Maven Deployment
	2min 23s	54ms	5s	1min 45s	1min 25s	19s	1min 57s	33s	43s
#11 Jul 11 23:54 2 commits	683ms	58ms	4s	1min 38s	2min 23s	650ms	3min 11s	55ms	1min 18s
#10 Jul 07 21:54 2 commits	20s	62ms	4s	2min 10s	1min 13s	50ms	3min 8s	56ms	1min 26s
#9 Jul 07 01:54 1 commits	996ms	67ms	4s	2min 6s	1min 8s	117ms	3min 16s	522ms	1min 24s



Service Status Details For Host **ci.pok.stglabs.ibm.com**

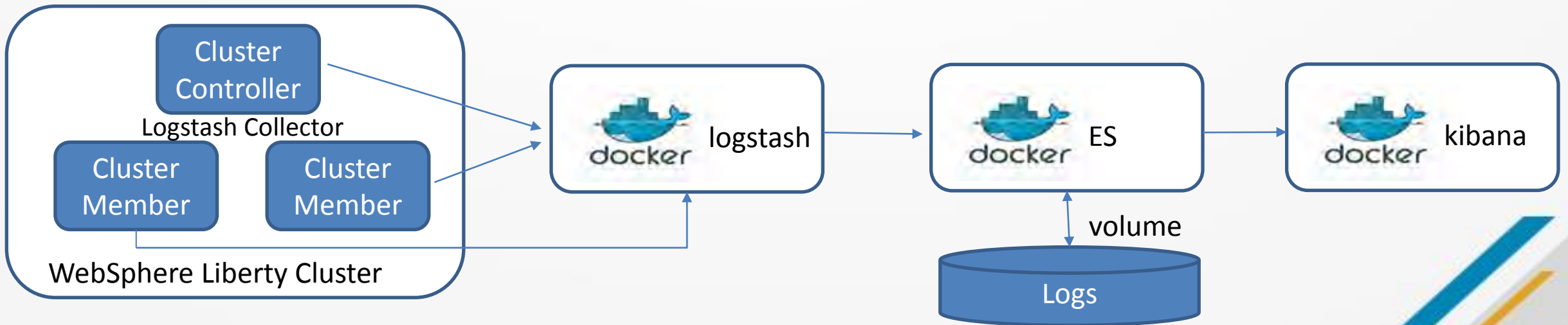
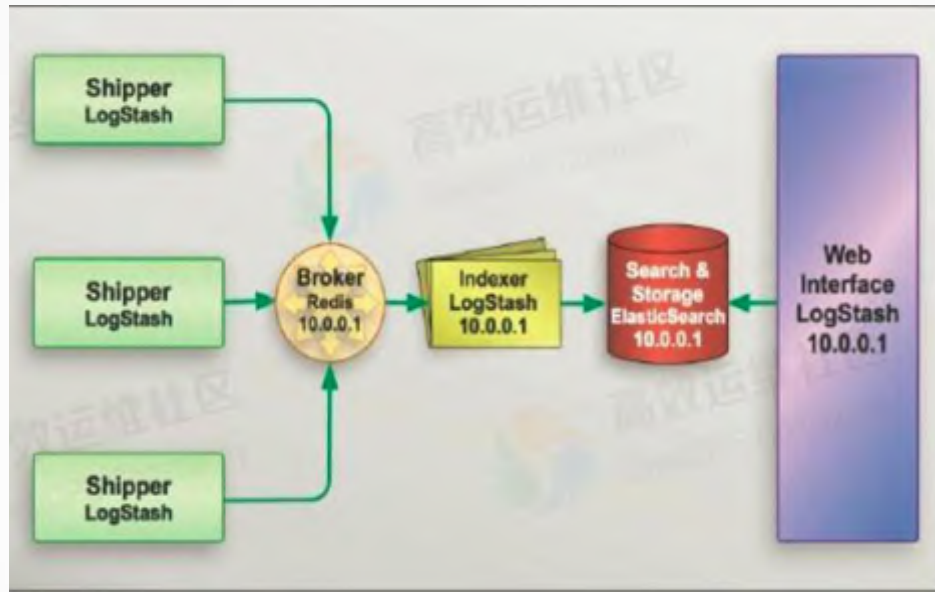
Limit Results: All

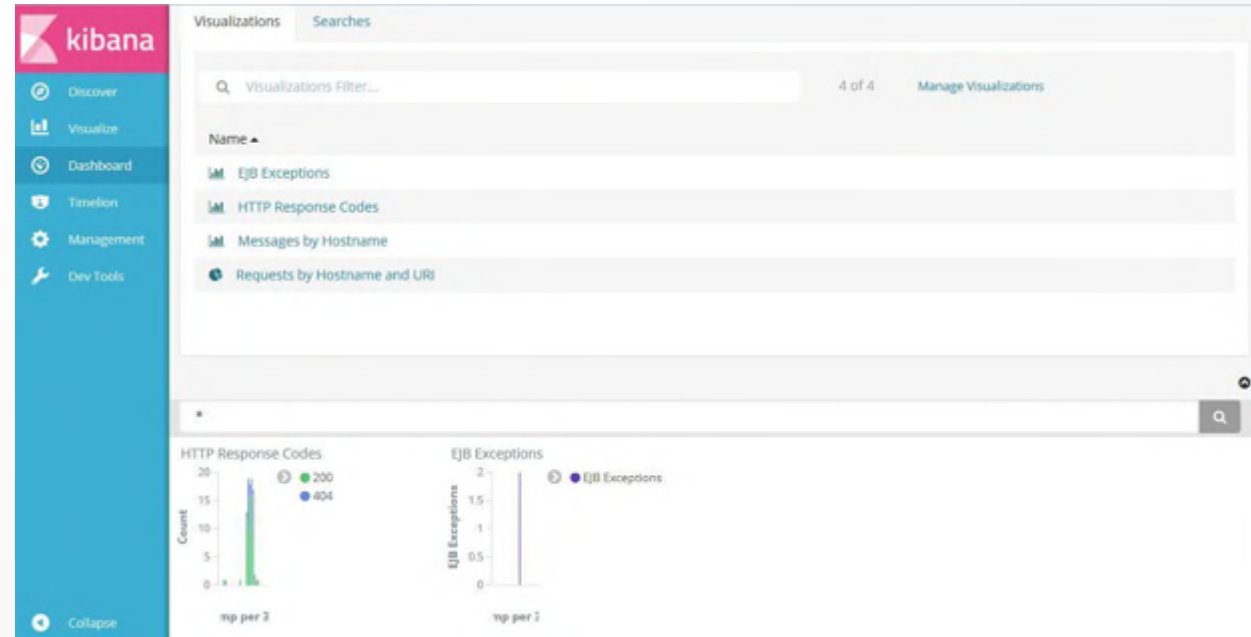
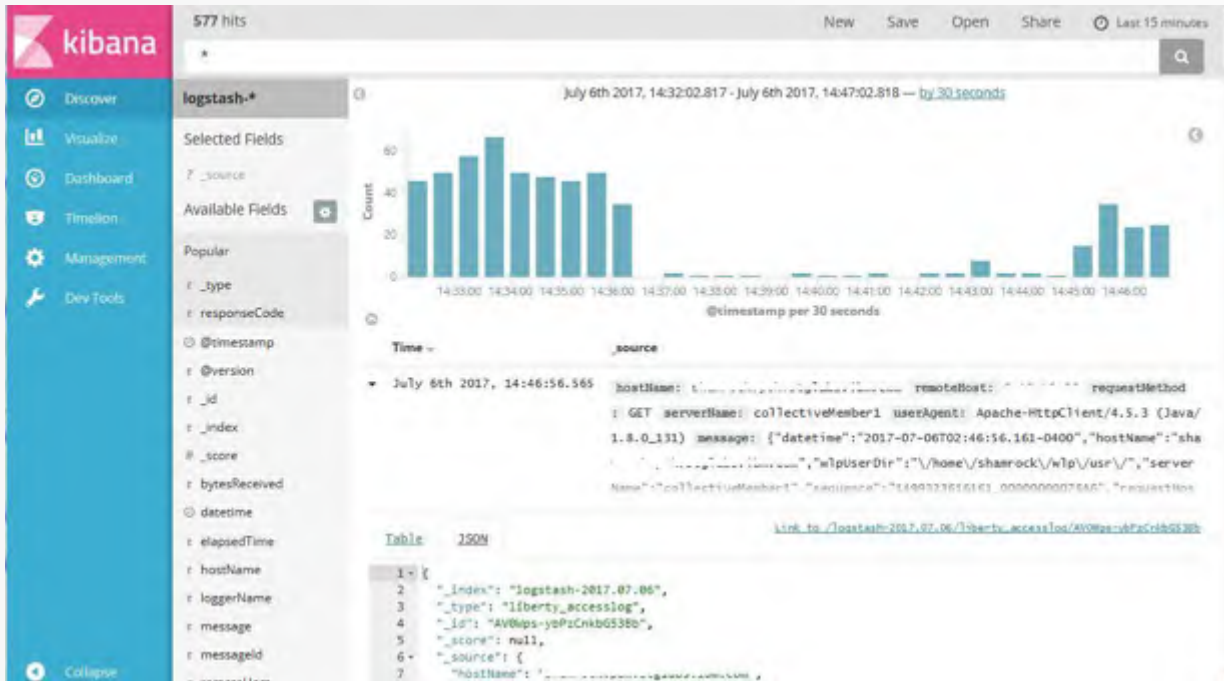
Host	Service	Status	Last Check	Duration	Attempt	Status Information
ci.pok.stglabs.ibm.com	COCKPIT	OK	07-03-2017 02:34:18	122d 15h 51m 2s	1/3	HTTP OK: HTTP/1.1 200 OK - 42340 bytes in 0.091 second response time
ci.pok.stglabs.ibm.com	HTTP	OK	07-03-2017 02:29:35	9d 15h 37m 16s	1/3	HTTP OK: HTTP/1.1 200 OK - 56050 bytes in 0.114 second response time
ci.pok.stglabs.ibm.com	Local: CPU Load	OK	07-03-2017 02:33:12	122d 15h 51m 36s	1/3	OK - load average: 0.01, 0.03, 0.05
ci.pok.stglabs.ibm.com	Local: System Disks	OK	07-03-2017 02:36:17	122d 15h 53m 31s	1/3	DISK OK - free space: / 42574 MB (43.50% inode=100%);
ci.pok.stglabs.ibm.com	Local: System Processes	OK	07-03-2017 02:33:56	122d 15h 51m 0s	1/3	PROCS OK: 200 processes with STATE = RSZDT
ci.pok.stglabs.ibm.com	Local: System Swap	OK	07-03-2017 02:33:59	122d 15h 48m 29s	1/3	SWAP OK - 99% free (3920 MB out of 3967 MB)
ci.pok.stglabs.ibm.com	Local: System Users	OK	07-03-2017 02:36:03	122d 15h 53m 12s	1/3	USERS OK - 2 users currently logged in
ci.pok.stglabs.ibm.com	PING	OK	07-03-2017 02:34:27	122d 15h 53m 29s	1/3	OK - ci.pok.stglabs.ibm.com: rta 0.192ms, lost 0%
ci.pok.stglabs.ibm.com	SSH	OK	07-03-2017 02:34:15	122d 15h 50m 58s	1/3	SSH OK - OpenSSH_6.6.1 (protocol 2.0)

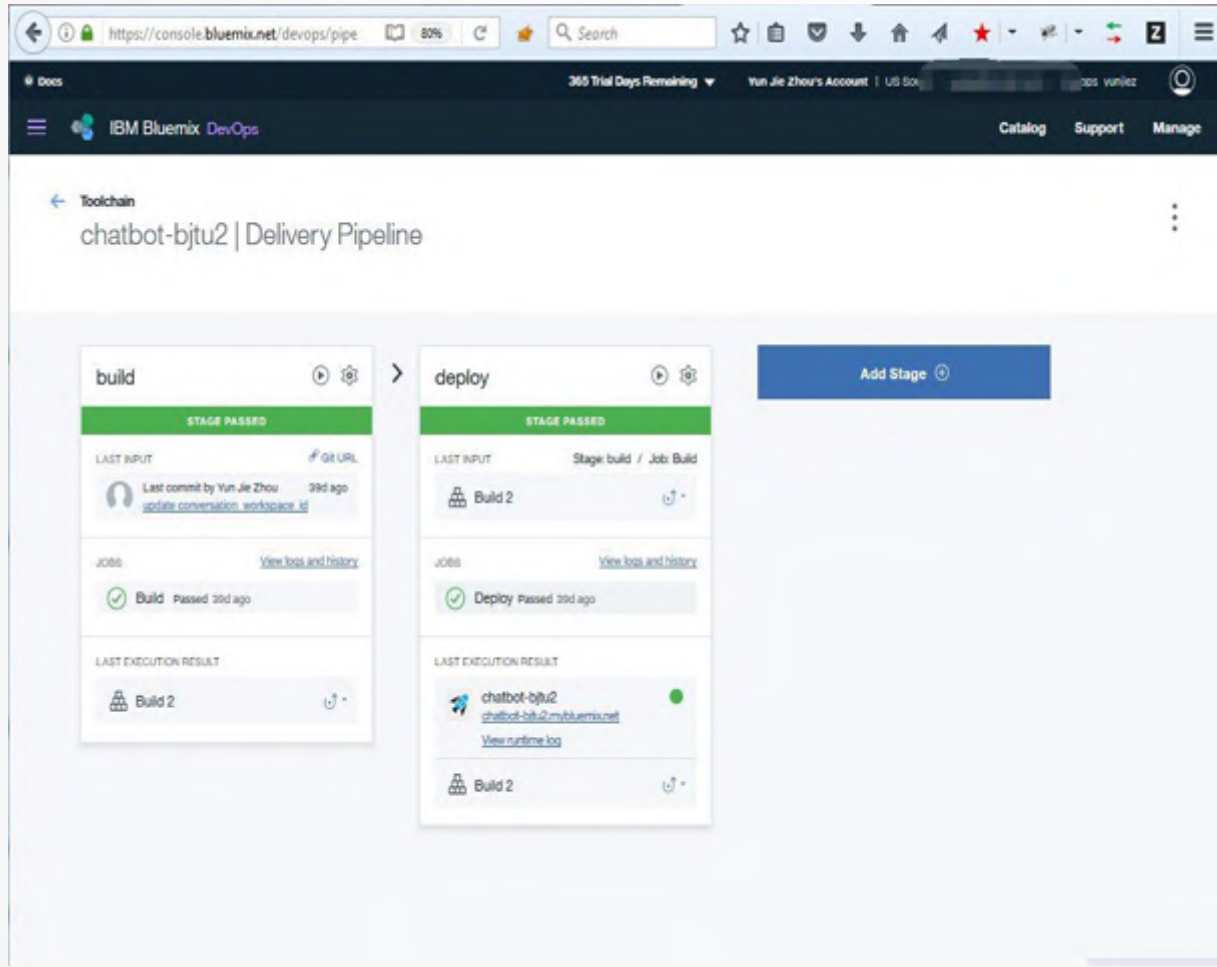
- Network Monitoring
- Server Monitoring
- Application Monitoring

Host State Breakdowns:

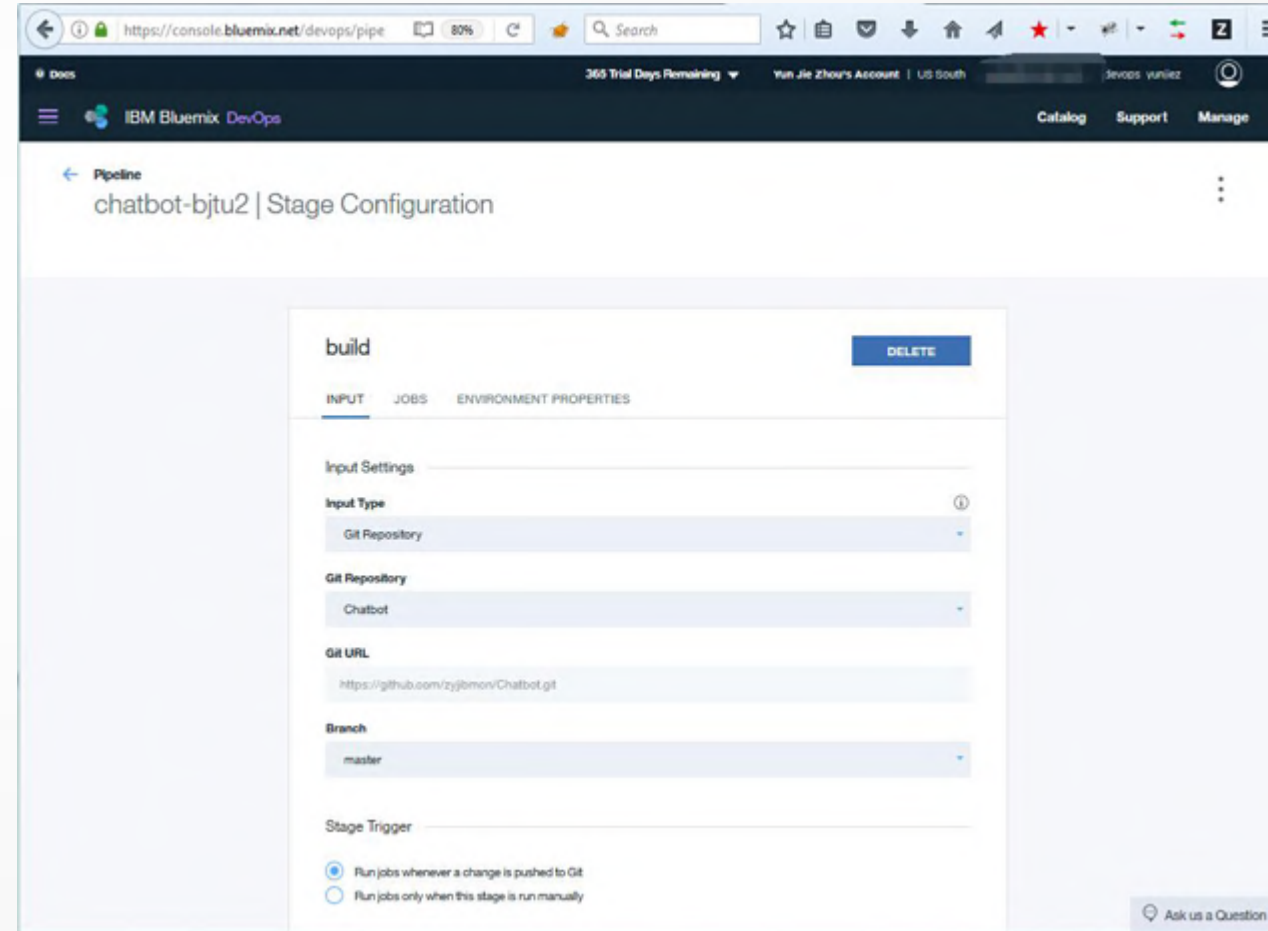
State	Type / Reason	Time	% Total Time	% Known Time
UP	Unscheduled	7d 0h 0m 0s	100.000%	100.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	7d 0h 0m 0s	100.000%	100.000%
DOWN	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
UNREACHABLE	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	0d 0h 0m 0s	0.000%	
	Total	0d 0h 0m 0s	0.000%	
All	Total	7d 0h 0m 0s	100.000%	100.000%



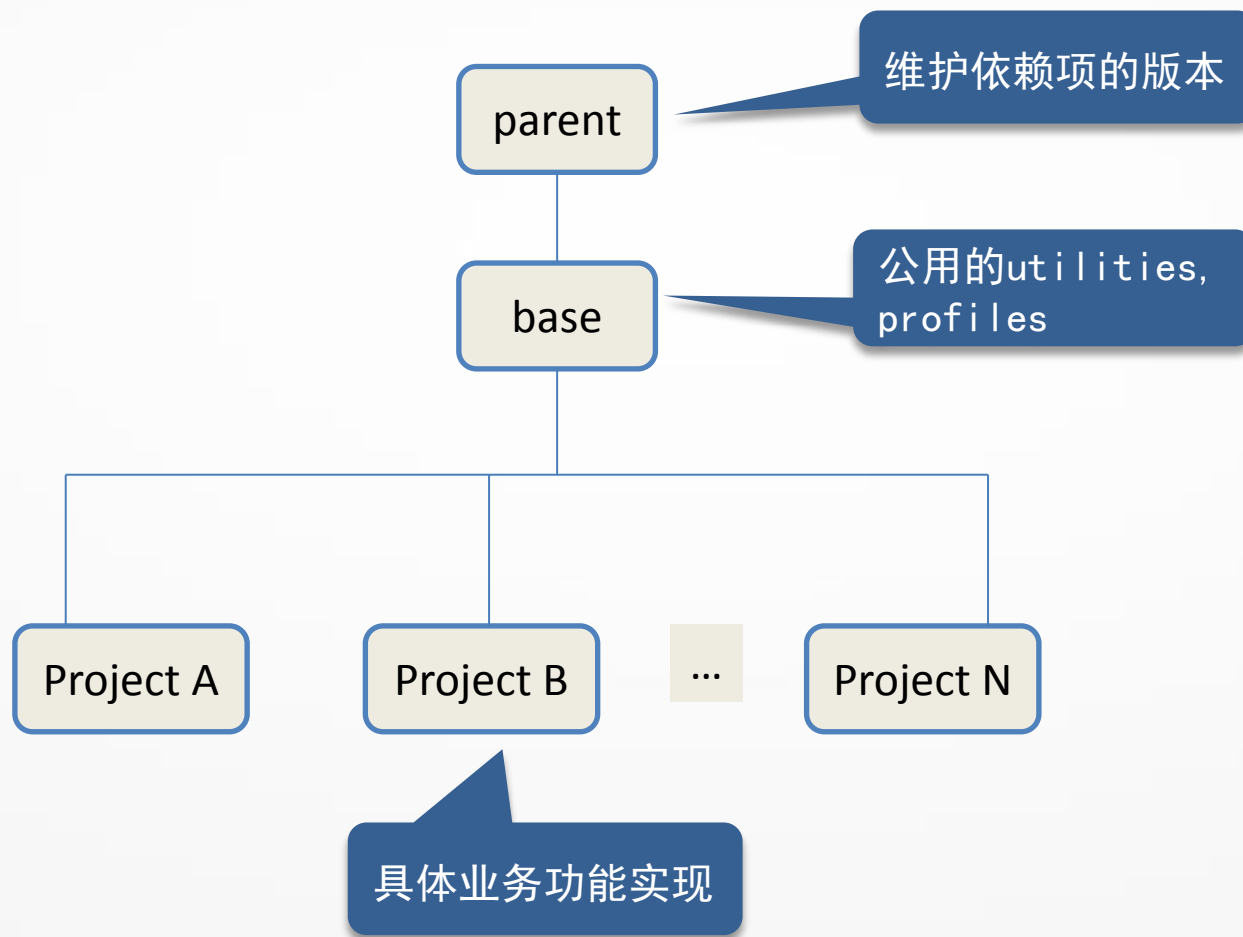




The screenshot shows the IBM Bluemix DevOps console interface. The breadcrumb navigation indicates the path: Toolchain > chatbot-bjt2 | Delivery Pipeline. The main content area displays two stage cards: 'build' and 'deploy', both with a green 'STAGE PASSED' header. The 'build' stage shows a last input from 'Yun.Jie.Zhou' and a job 'Build' that passed 29d ago. The 'deploy' stage shows a last input 'Stage build / Job Build' and a job 'Deploy' that passed 29d ago. Below the jobs, the 'LAST EXECUTION RESULT' for each stage is shown, with 'Build 2' and 'chatbot-bjt2' respectively. An 'Add Stage' button is visible on the right side of the stage cards.

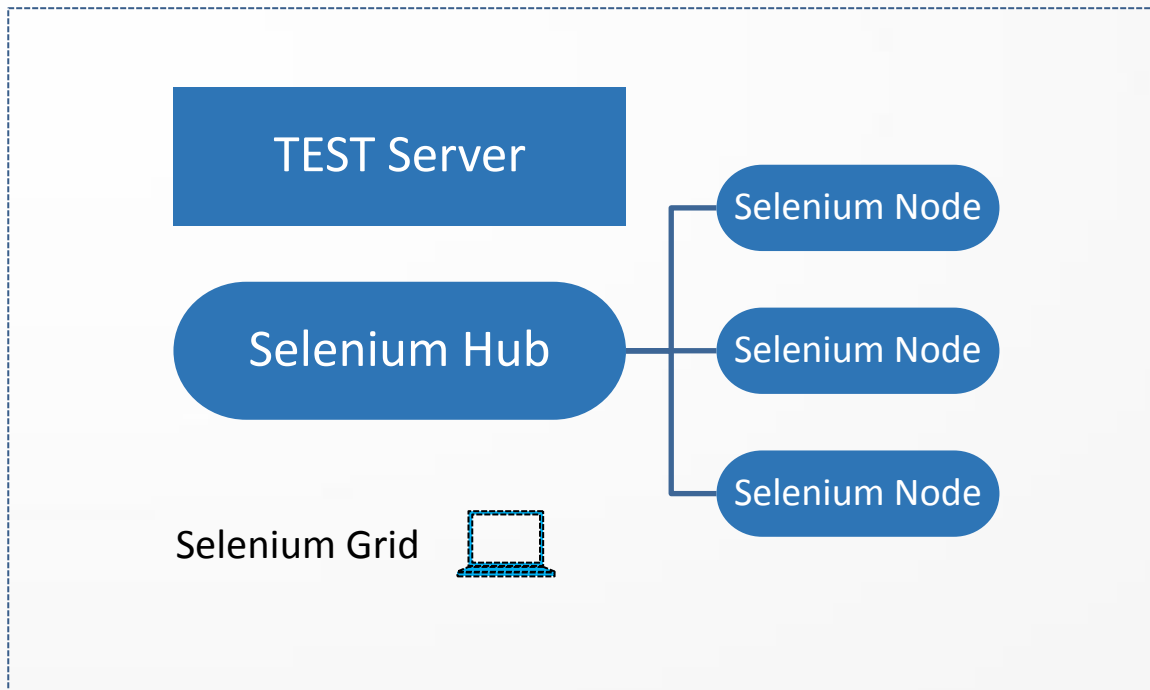


The screenshot shows the 'Stage Configuration' page for the 'build' stage in the 'chatbot-bjt2' pipeline. The breadcrumb navigation is Pipeline > chatbot-bjt2 | Stage Configuration. The 'build' stage configuration is displayed with a 'DELETE' button in the top right corner. The configuration is organized into sections: 'INPUT', 'JOBS', and 'ENVIRONMENT PROPERTIES'. Under 'INPUT SETTINGS', the 'Input Type' is set to 'Git Repository', the 'Git Repository' is 'Chatbot', and the 'Git URL' is 'https://github.com/zyj6mov/Chatbot.git'. The 'Branch' is set to 'master'. Under 'STAGE TRIGGER', the 'Run jobs whenever a change is pushed to Git' option is selected.



多机环境: Selenium Grid

单机环境: Headless Selenium



- CI/CD/DevOps
- 使用Jenkins 2 Pipeline 定义CI/CD pipeline
- 案例： 测试工具开发中CI/CD的落地