

泰康DevOps转型之Pipeline实践之路

衣后锋

泰康云DevOps平台架构师

2018 中国·上海





1 保险企业新业态的软件交付需求和挑战

2 DevOps工具链与Pipeline

3 Pipeline as Code

4 泰康云DevOps平台实践与未来展望



数字化浪潮的兴起和深化，保险业开始面临巨大压力。伴随互联网、大数据，以及人工智能等技术的发展，新业务形态、新业务需求乃至新业务创新都对现有IT提出了新的挑战。

泰康保险集团首次跻身2018年《财富》世界500强排行榜，标志着泰康迈入全球大型保险金融服务集团的行列。未来，打造“保险+医疗养老”生态闭环，引领服务业和供给侧改革，助力民生发展，服务经济社会。

建设**最大健康云**，全力支撑泰康生态体系



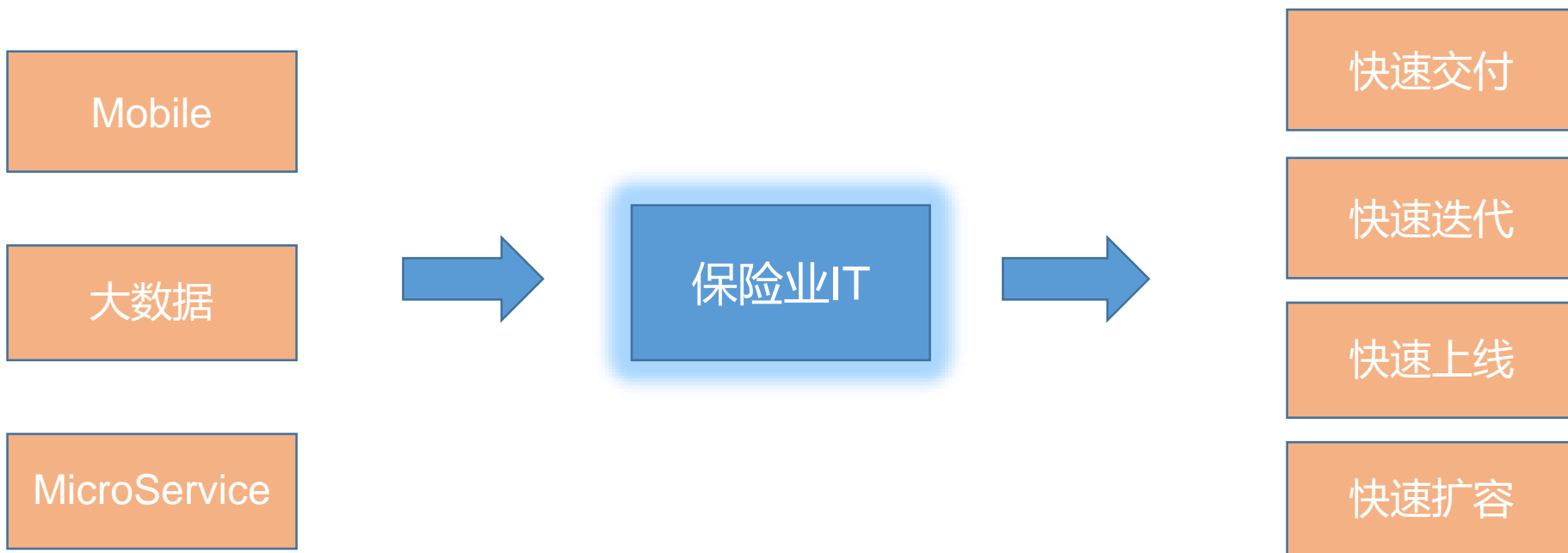
泰康
Taikang

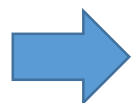


泰康保险集团 世界500强企业



业务发展带来的技术挑战





1 保险企业新业态的软件交付需求和挑战

2 DevOps工具链与Pipeline

3 Pipeline as Code

4 泰康云DevOps平台实践与未来展望



DevOps之Pipeline

- Pipeline流水线是指软件从版本控制库到用户手中这一过程的自动化实现是持续交付与DevOps的 **核心工程实践**；
- Pipeline流水线的自动化和持续流动，才能保证在不同阶段、不同节点上产品发布的一致性和稳定性，同时，也才能 **消除由于人工操作所引入的人为风险**，同时提高效率， **消除“等待”与“浪费”**。



DevOps之标准交付流程



基本信息

- 名称
- 负责人
- 备注信息
- ...



构建信息

- 开发语言
- 开发框架
- 代码库地址
- ...



集群管理

- 开发环境
- 测试环境
- 生产环境
- ...



网络资源

- 负载均衡
- 内网域名
- 公网域名
- ...



部署资源

- 程序包
- 配置包
- 文件包
- ...

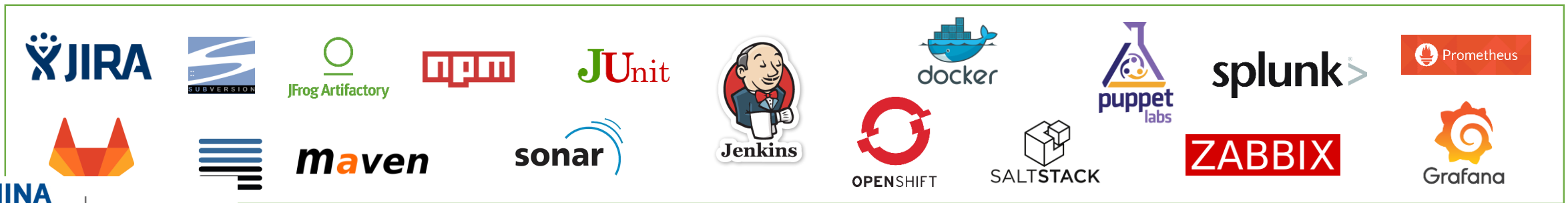
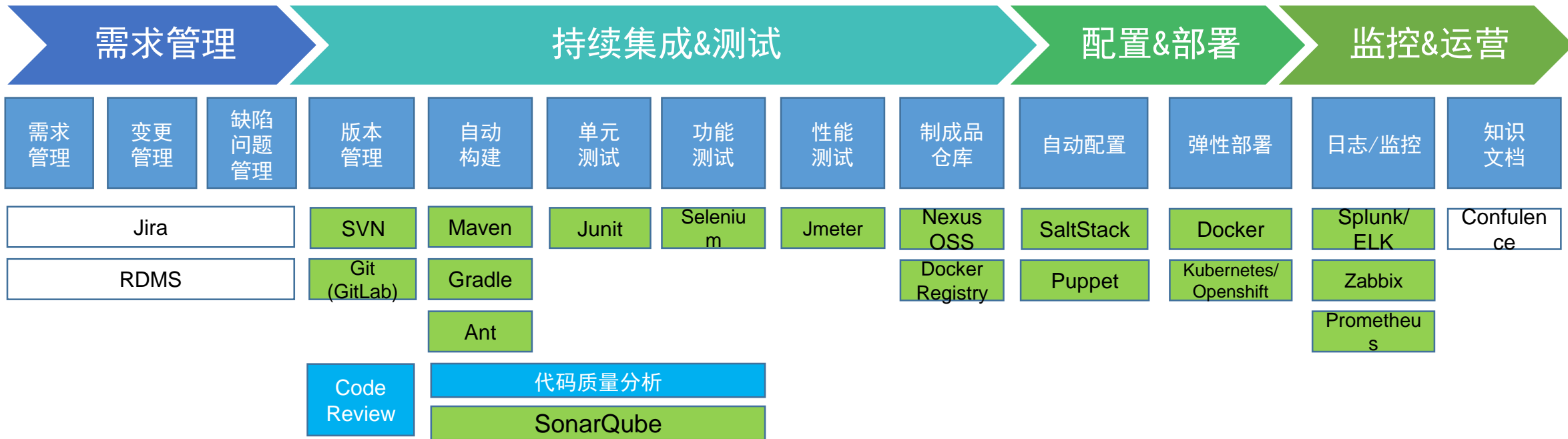


环境依赖

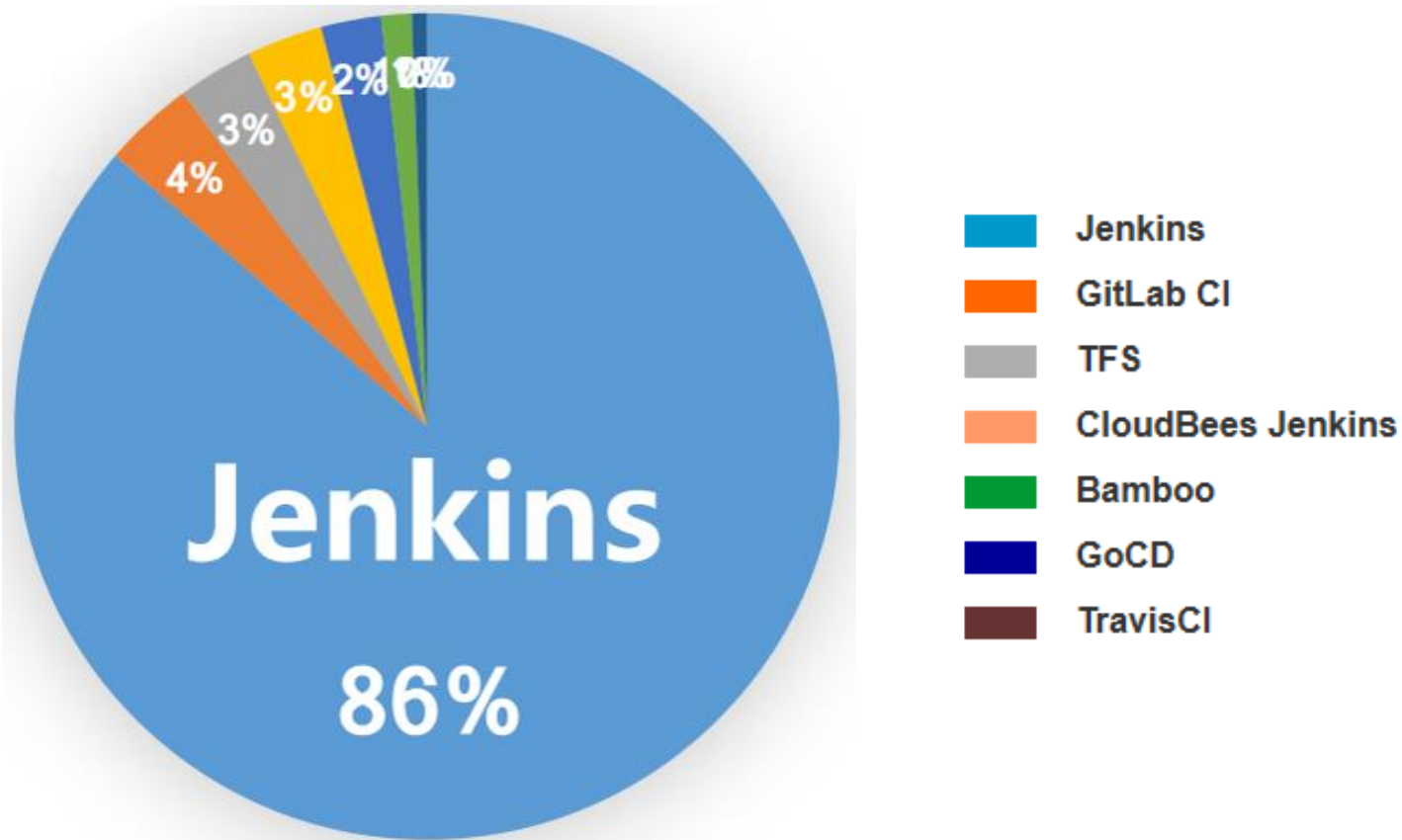
- 基础镜像
- 中间件
- ...



基于开源生态打造DevOps工具链

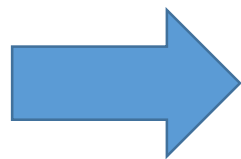


Jenkins Pipeline 端到端流水线



来源：2017年中国IT企业 DevOps 持续交付流水线现状调查报告





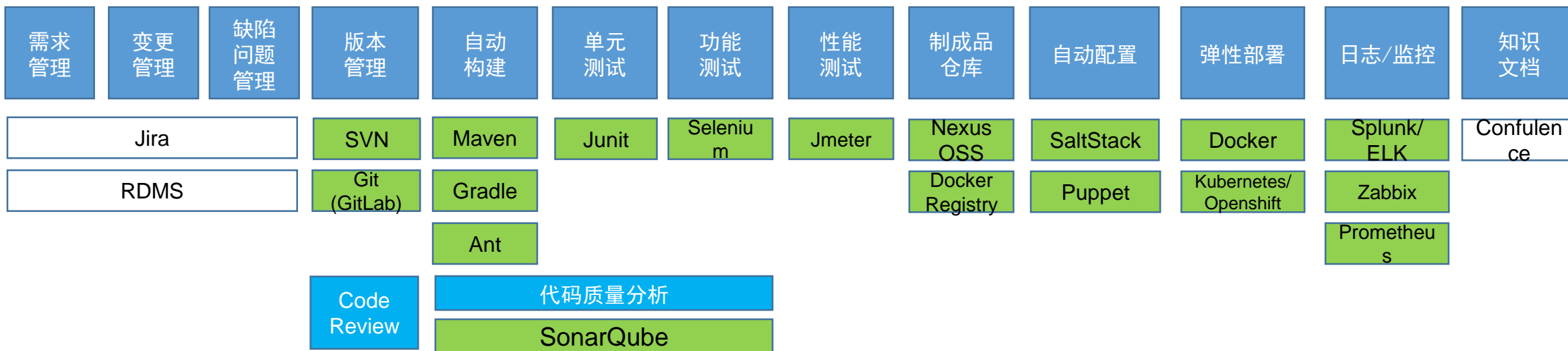
Jenkins 2.0 Pipeline

需求管理

持续集成&测试

配置&部署

监控&运营



自动集成

Jenkins (Pipeline)



Pipeline落地填坑指南

- Pipeline + 标准交付流程：一软一硬，都不能放松；
- 当企业想要迁移到一个 DevOps 模型时，经常需要偿还高等级的技术债务：平衡短期收益与长期收益；
- 循序渐进：降低学习曲线，培养用户习惯。



1 保险企业新业态的软件交付需求和挑战

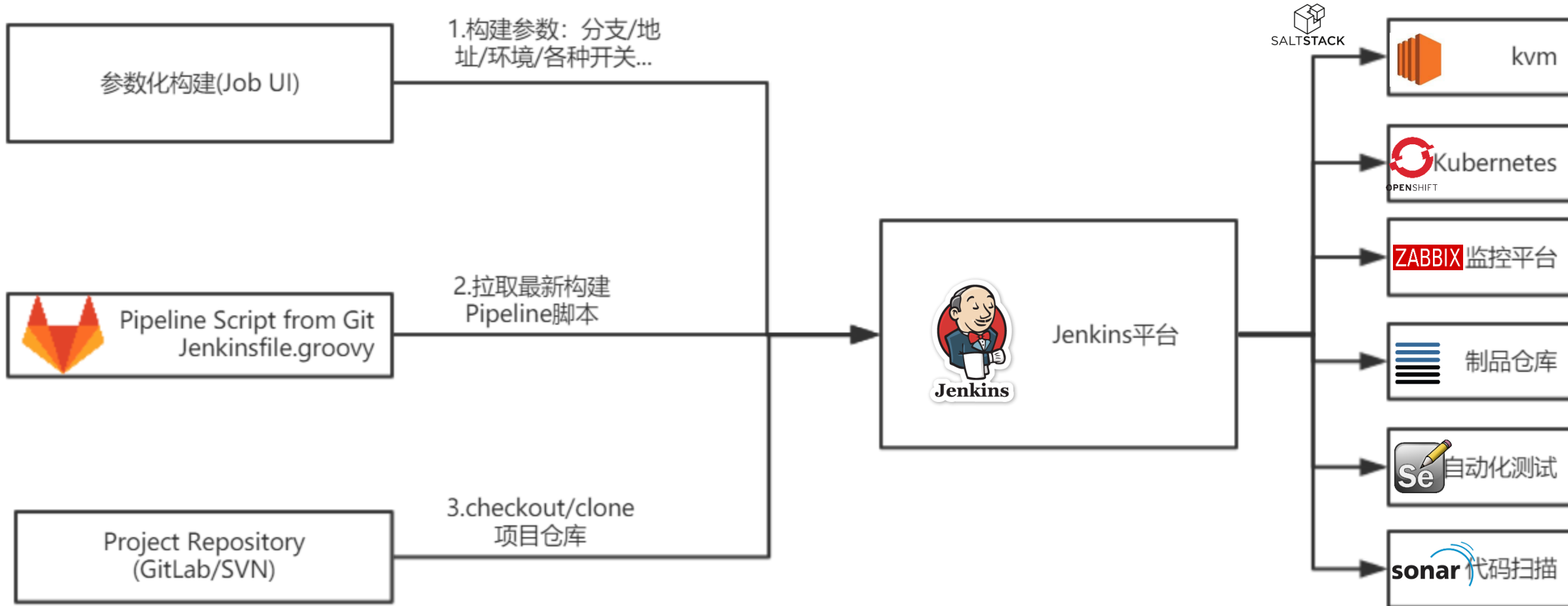
2 DevOps工具链与Pipeline



3 Pipeline as Code

4 泰康云DevOps平台实践与未来展望





Pipeline设计原则：代码化(Pipeline as Code)

由Groovy DSL编写
声明式简化代码；
脚本式做逻辑控制；

特点：

- 高扩展性
- 灵活插拔
- 集成成本低

```

1  #!/usr/bin/groovy
2  import groovy.json.*
3  import hudson.model.*
4  @NonCPS
5
6  // 代码库信息
7  String srcUrl = "${env.srcUrl}"
8  String srcType = "${env.srcType}"
9  String tagName = "${env.tagName}"
10 String branchName = "${env.branchName}"
11 String moduleName = "${env.moduleName}"
12 String credentialsId = "${env.credentialsId}"
13
14 // 服务信息
15 String serviceName = "${env.serviceName}"
16 String targetHosts = "${env.targetHosts}"
17 String targetDir = "${env.targetDir}/${serviceName}"
18 String workspace = "${env.workspace}/${JOB_NAME}"
19 String activeENV = "${env.activeENV}"
20 String user = "${env.user}"
21 String port = "${env.port}"
22 String livenessUrl = "${env.livenessUrl}"
23 String classPath = "${env.classPath}"
24 String javaOption = "${env.javaOption}"
25 String startType = "${env.startType}"
26 String javaVersion = "${env.javaVersion}"
27 String dependency = "${env.dependency}"
28
29 //Jenkins 信息
30 String nodeSlave = "${env.nodeSlave}"
31 String skipDeploy = "${env.skipDeploy}"
32 String skipSonar = "${env.skipSonar}"
33 String buildType = "${env.buildType}"
34 String buildShell = "${env.buildShell}"
35 String opsService = "${env.opsService}"

```

变量初始化

```

79 //Json初始化
80 def JsonInto(value){
81     def prettyJson = JsonOutput.prettyPrint(value)
82     def jsonSlurper = new JsonSlurper()
83     def jsonSaltresult = jsonSlurper.parseText(prettyJson)
84     //println(jsonSaltresult)
85     def result
86     for (host in hosts){
87         result = jsonSaltresult[0][host]
88         println("${host}" + '-->' + "${result}")
89     }
90     return result
91 }
92
93 // 主机格式化
94 def HostInit(targetHosts){
95     hostList = [ ]
96     if (length < 2){
97         String value = hosts[0]
98         hostList.add(value)
99     } else {
100        for (i = 0; i < length-1; i+=2 ){
101            String value = hosts[i] + ',' + hosts[i+1]
102            hostList.add(value)
103        }
104    }
105    return hostList
106 }

```

数据格式化

Pipeline设计原则：模块化/原子化

代码检出

```

120 // 代码检出
121 def GetCode(srcType,srcUrl,tagName,branchName,credentials)
122     if(srcType == "SVN") {
123         if(tagName == "null") {
124             svnUrl = "${srcUrl}@HEAD"
125         }else{
126             svnUrl = "${srcUrl}/${tagName}@HEAD"
127         }
128         checkout([
129             $class: 'SubversionSCM',
130             additionalCredentials: [],
131             filterChangelog: false,
132             ignoreDirPropChanges: false,
133             locations: [[credentialsId: "${credentialsId}"],
134             depthOption: 'infinity',
135             ignoreExternalsOption: true,
136             remote: "${svnUrl}"]],
137             workspaceUpdater: [$class: 'CheckoutUpdater']]
138     )
139     } else if(srcType == "Git") {
140         //delete 'origin/'
141         if (branchName.startsWith('origin/')){
142             branchName=branchName.minus("origin/")
143         }
144
145         if(tagName == "null"){
146             pathName = "*/${branchName}"
147         }else{
148             pathName = "refs/tags/${tagName}"
149         }
150         checkout([$class: 'GitSCM', branches: [[name: "${pathName}"]],
151             doGenerateSubmoduleConfigurations: false,
152             extensions: [], submoduleCfg: [],
153             userRemoteConfigs: [[credentialsId: "${credentialsId}",

```

Build

```

180 // 构建 [mvn/ant]
181 def Build(javaVersion,buildType,buildDir,buildShell){
182     if (buildType == 'maven'){
183         Home = tool 'M3'
184         buildHome = "${Home}/bin/mvn"
185     } else if (buildType == 'ant'){
186         Home = tool 'ANT'
187         buildHome = "${Home}/bin/ant"
188     } else{
189         error 'buildType Error [maven|ant]'
190     }
191     echo "BUILD_HOME: ${buildHome}"
192     //判断
193     if (javaVersion == 'jdk7') {
194         println("---USE JDK7---")
195         sh """
196             export JAVA_HOME=/usr/local/jdk1.7.0_79
197             export PATH=\$JAVA_HOME/bin:\$PATH
198             export CLASSPATH=.\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar
199             java -version
200             cd ${buildDir} && ${buildHome} ${buildShell}
201             """
202     } else if (javaVersion == 'jdk6') {
203         println("---USE JDK6---")
204         sh """
205             export JAVA_HOME=/usr/local/jdk1.6.0_45
206             export PATH=\$JAVA_HOME/bin:\$PATH
207             export CLASSPATH=.\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar
208             java -version
209             cd ${buildDir} && ${buildHome} ${buildShell}
210             """
211     } else {
212         println("---USE JDK8---")
213         sh """
214             export JAVA_HOME=/usr/java/jdk1.8.0_111
215             export PATH=\$JAVA_HOME/bin:\$PATH
216             export CLASSPATH=.\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar
217             java -version
218
219             cd ${buildDir} && ${buildHome} ${buildShell}
220             """
221     }
}

```


Pipeline设计原则：模块化/原子化

Build

```

180 //构建 [mvn/ant]
181 def Build(javaVersion,buildType,buildDir,buildShell){
182     if (buildType == 'maven'){
183         Home = tool 'M3'
184         buildHome = "${Home}/bin/mvn"
185     } else if (buildType == 'ant'){
186         Home = tool 'ANT'
187         buildHome = "${Home}/bin/ant"
188     } else{
189         error 'buildType Error [maven|ant]'
190     }
191     echo "BUILD_HOME: ${buildHome}"
192     //判断
193     if (javaVersion == 'jdk7') {
194         println("---USE JDK7---")
195         sh """
196             export JAVA_HOME=/usr/local/jdk1.7.0_79
197             export PATH=\$JAVA_HOME/bin:\$PATH
198             export CLASSPATH=.\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar
199             java -version
200             cd ${buildDir} && ${buildHome} ${buildShell}
201             """
202     } else if (javaVersion == 'jdk6') {
203         println("---USE JDK6---")
204         sh """
205             export JAVA_HOME=/usr/local/jdk1.6.0_45
206             export PATH=\$JAVA_HOME/bin:\$PATH
207             export CLASSPATH=.\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar
208             java -version
209             cd ${buildDir} && ${buildHome} ${buildShell}
210             """
211     } else {
212         println("---USE JDK8---")
213         sh """
214             export JAVA_HOME=/usr/java/jdk1.8.0_111
215             export PATH=\$JAVA_HOME/bin:\$PATH
216             export CLASSPATH=.\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar
217             java -version

```

```

145 // 进程检查
146 def CheckService(Hosts,port,livenessUrl) {
147     println "发布主机信息: " + ' |' + "Hosts:" + Hosts
148     try {
149         if(livenessUrl != "null"){
150             sh "salt -C L@\"${Hosts}\" cmd.run 'curl http://127.0.0.1:${port}/${livenessUrl}'";
151         }
152     } catch(ERROR) {
153         currentBuild.result = 'FAILURE'
154         echo "*****${Hosts}主机服务进程检查结束*****"
155     }
156 }

```

发布验证

```

271 //代码扫描
272 def SonarScan(skipSonar,serviceName,srcDir){
273     try {
274         def scanHome = tool 'Scanner2.6'
275         sh "cd ${srcDir} && ${scanHome}/bin/sonar-scanner -Dsonar.projectName=${
276     } catch (e){
277         error 'skipSonar [false|true] '
278     }
279 }

```

代码扫描

```

217             java -version
                cd ${buildDir} && ${buildHome} ${buildShell}

```

- 功能开关：
- 分支/Tag获取
- 主机选择
- 构建类型
- 代码检查
- 部署
- 发布验证

```
// Triggers
SKIP_DOCKER = false
SKIP_KUBE = false
SKIP_COMPILE = false
SKIP_GIT = false
SKIP_TEST = false
SKIP_SONAR = true
SKIP_REPORT = false
```

需要如下参数用于构建项目：

serviceName

服务名,放置jar包的目录

branchName

2.0.1
2.0.4
1.0.0
2.0.5

If you selected revisions as a type of presented data and working space is empty, the plug needs slow connection or repository is large.

srcUrl

targetHosts t0-...-eb01
 t1-...-eb02
 t2-...-eb03
 t3-...-eb04

opsService

服务操作: 启动/停止/查看状态

moduleName

mavenShell

srcType

代码库类型

targetDir

目标主机路径

port

activeENV

启动环境

user

启动用户

credentialsId

classPath

workspace

jenkins工作目录

nodeSlave

开始构建

Pipeline设计原则：代码化(Pipeline script from SCM)

Pipeline设计原则
代码化(Pipeline script from SCM)

Pipeline

Definition **Pipeline script from SCM**

SCM **Git**

Repositories

Repository URL **ssh://git@git.██████████.st/erp-██████████-android.git**

Credentials **root** **Add**

高级...

Add Repository

Branches to build

Branch Specifier (blank for 'any') ***/master**

X

Add Branch

Git executable **Default**

源码库浏览器 **(自动)**

Additional Behaviours **Add**

Script Path **jenkins/build_1.groovy**

Lightweight checkout

在这里配置构建脚本的仓库、分支和文件

签出指定分支或tag的代码

Git Parameter

Name:

Description:

Parameter Type:

Branch:

Branch Filter:

Tag Filter:

Sort Mode:

Default Value:

Selected Value:

Use repository:

Quick Filter:

List Subversion tags (and more)

Name:

Repository URL:

Credentials:

Tags filter:

Default value:

Maximum tags to display:

Sort newest first

Sort Z to A



Jenkins基于角色的访问控制策略

1. 用户接入 [LDAP Plugin](#)

2. 项目/角色/矩阵权限控制 [Matrix Authorization Strategy Plugin](#)

系统管理 -> 全局安全配置 -> 访问控制 -> 授权策略 -> Role-Based Strategy

定义正则匹配:

Roles myservice1 ^myk8s1-.*

Roles myservice2 ^myk8s2-.*

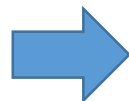
Project roles

Role	Pattern	Credentials				Gerrit		任务									
		Create	Delete	ManageDomains	Update	View	ManualTrigger	Retrigger	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Write
<input checked="" type="checkbox"/> CAP	^CAP\$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> PLU	^PLU\$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> SC	^SC\$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> T	^T\$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> U	^U\$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> W	^W\$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> R	^R\$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

1 保险企业新业态的软件交付需求和挑战

2 DevOps工具链与Pipeline

3 Pipeline as Code



4 泰康云DevOps平台实践与未来展望



原生Jenkins Pipeline不足

- DevOps度量无法做到信息统一收集展示
- 用户使用无法闭环：需在各个工具平台切换
- 权限控制太弱，无法满足企业级需求
- 各个工具链平台无法有效整合，运维管理成本高



DevOps统一管理平台建设目标

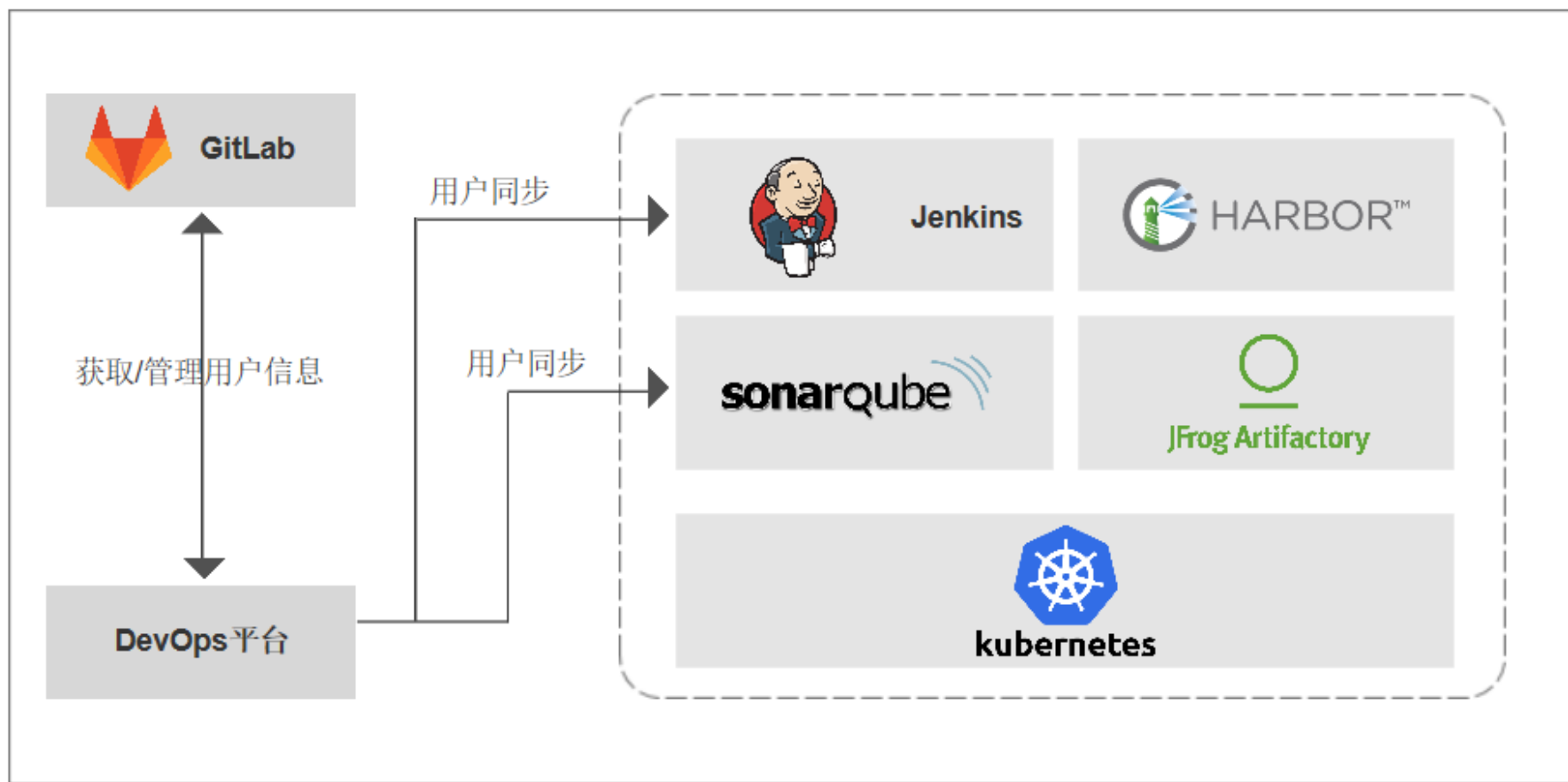
- 自服务化与易用性
- 提供最佳实践
- 灵活集成其他第三方平台，降低工具链整体运维成本
- 以Jenkins和GitLab为核心构建DevOps平台



DevOps统一管理平台



用户管理



平台最佳实践：CodePipeline命名规则

应用命名：业务-应用-类型

示例：

service-monitor-service

service-monitor-web

CodePipeline命名

示例：

service-cus-service_DEV

service-app-ui_STAG

||| 应用发布 / 新建导航

* 选择业务 * 选择应用 没有找到应用? [维护](#)

* 选择环境 开发环境(development)DEV 测试环境(testing)TEST 生产环境(production)PRD 模拟环境(Staging Server)STAG

用户验收测试环境(User Acceptance Testing)UAT 培训环境(training)TRN

项目(pipeline)名称: tcmp-devops-web_DEV

构建一个JAVA的软件项目
基于Maven/Ant来构建你的java体系风格的项目

构建一个NodeJs的软件项目
基于npm来构建你的前端体系风格的项目

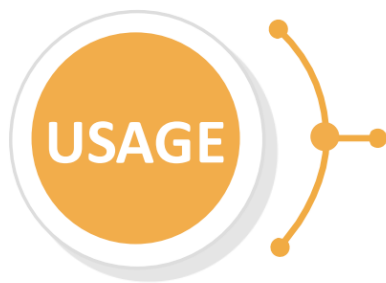
构建一个Nginx的软件项目
基于Nginx来构建你的前端体系风格的项目



新建

job名称	应用名称	申请人	最近构建	构建结果	审核状态	操作
tcrap-cmdb-service_DEV	tcrap-cmdb-service		2018-08-03 15:02:57	成功	通过	构建 跳转 删除
tcrap-monitor-service_TEST	tcrap-monitor-service		2018-08-10 10:25:38	成功	通过	构建 跳转 删除
tcrap-monitor-service_STAG	tcrap-monitor-service		2018-07-05 10:40:57	成功	通过	构建 跳转 删除
tcrap-monitor-service_DEV	tcrap-monitor-service		2018-08-10 22:53:44	成功	通过	构建 跳转 删除
tcrap-monitor-web_TEST	tcrap-monitor-web		2018-08-09 17:20:13	成功	通过	构建 跳转 删除
tcrap-monitor-web_STAG	tcrap-monitor-web		2018-08-08 17:10:43	成功	通过	构建 跳转 删除
tcrap-devops-service_TEST	tcrap-devops-service		2018-08-10 17:31:52	成功	通过	构建 跳转 删除
tcrap-devops-service_DEV	tcrap-devops-service		2018-08-10 22:52:44	成功	通过	构建 跳转 删除
tcrap-devops-service_STAG	tcrap-devops-service		2018-08-08 20:05:51	成功	通过	构建 跳转 删除
tcrap-gateway-service_DEV	tcrap-gateway-service		2018-08-10 22:57:51	成功	通过	构建 跳转 删除
tcrap-gateway-service_STAG	tcrap-gateway-service		2018-08-08 17:05:09	成功	通过	构建 跳转 删除
tcrap-gateway-service_TEST	tcrap-gateway-service		2018-08-07 18:34:05	成功	通过	构建 跳转 删除
tcrap-cmdb-service_DEV	tcrap-cmdb-service		2018-08-11 16:31:06	成功	通过	构建 跳转 删除

CodePipeline平台-当前进展



- 服务(应用)数量 300+
- Pipeline数量 1000+
- VM/容器数量 2000+
- 快速增长中

...



未来展望 V2.0

- 信息统一收集展示，软件生命周期全面数据分析
- 多渠道自动化监听及通知系统
- 快速可用和弹性扩展，提升资源利用率



THANKS

Website:
chinadevopsdays.org/

Global Website:
www.devopsdays.org/events/2018-shanghai/

Official Email:
organizers-shanghai-2018@devopsdays.org



Official Wechat

