

DevOps

chenshaowen

DevOps及工具链

目录

- 什么是 DevOps
- 敏捷开发
- DevOps 相关的工具
- 问答与实验

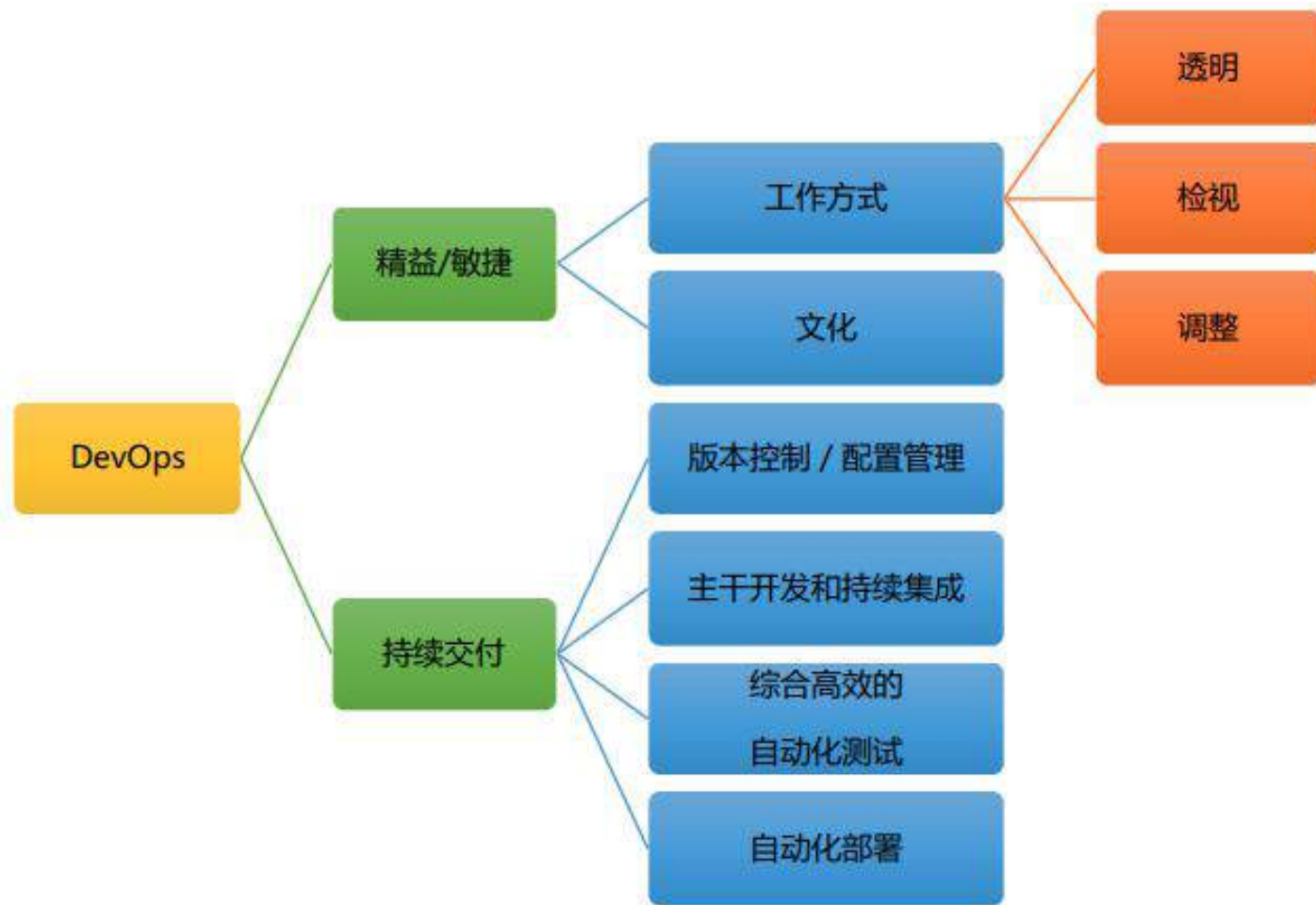
新的商业环境下对效率提出了更高要求

- 一个门户黄页就成就了搜狐
- 一个聊天工具就成就了腾讯
- 一个只有几个跳转页面的 hao123 都能卖几千万
- 一个 APP 就是一家公司
- 一个公众号、抖音号就能养活一帮人
- ToC 的生意越来越难，ToB 市场正在迅速增长

什么是 DevOps

- DevOps 是一种软件工程文化和实践，旨在统一整合软件开发和软件运维。
- DevOps 的特点是强烈倡导对构建软件的所有环节（从集成、测试、发布到部署和基础架构管理）进行全面的自动化和可视化。
- DevOps 的目标是缩短开发周期，提高部署频率和更可靠地发布，与业务目标保持一致。

DevOps 包含的内容



- 既要人的协作，也要工具支撑
- 既要流程规范，也要约束力

DevOps 文化建设

开发和IT运维更多的协作、沟通

提高整个生产系统工作流的速度



Embracing
a DevOps
Culture

减少工作移交和等待

团队自治

勇于创新

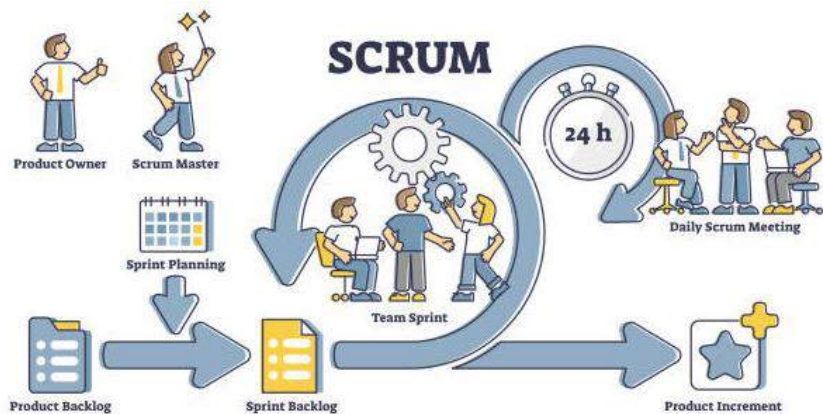
敢于冒险

高度信任

包容失败

大家喜欢在怎样的团队工作？

让信息流通起来



Scrum 迭代



项目看板



站立会议

传统开发模型

- 长周期
- 大开发
- 高风险



新的开发模型

- 拆解任务



- 迭代开发



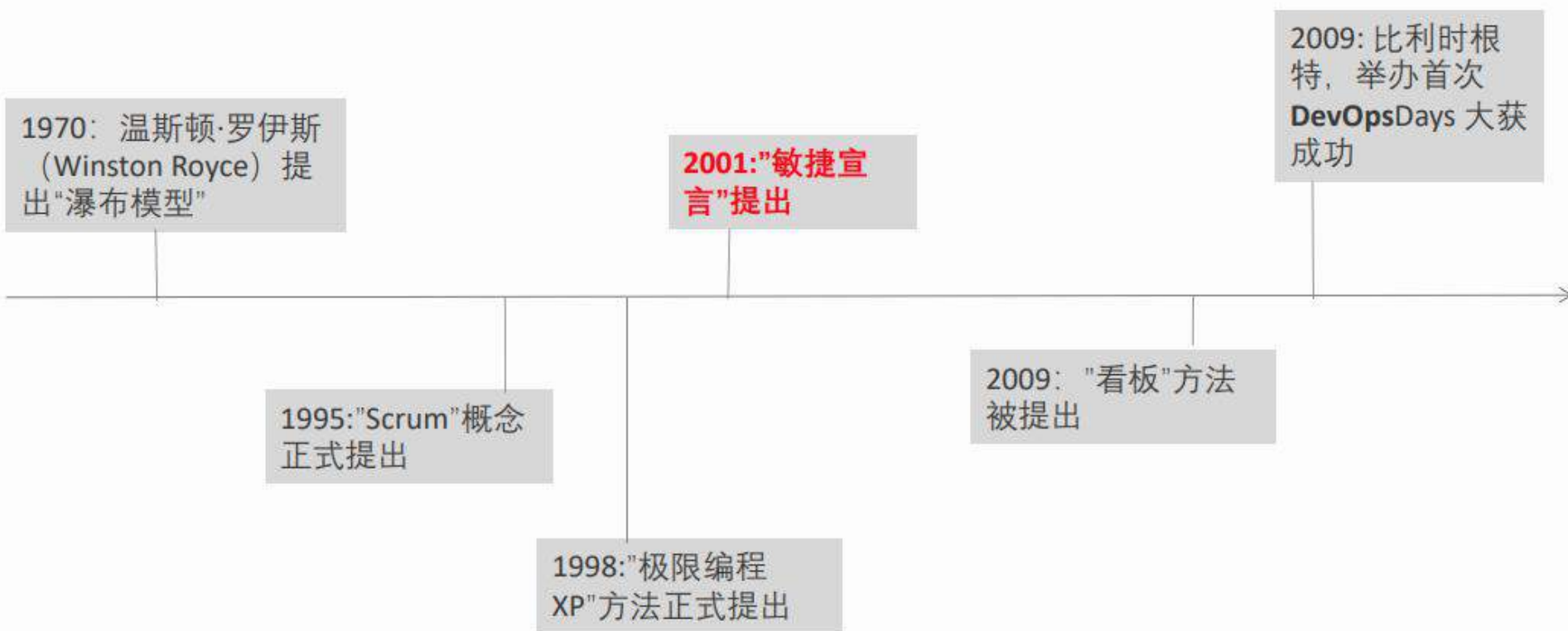
- 逐步改进

两种形态的交付，稳态和敏态

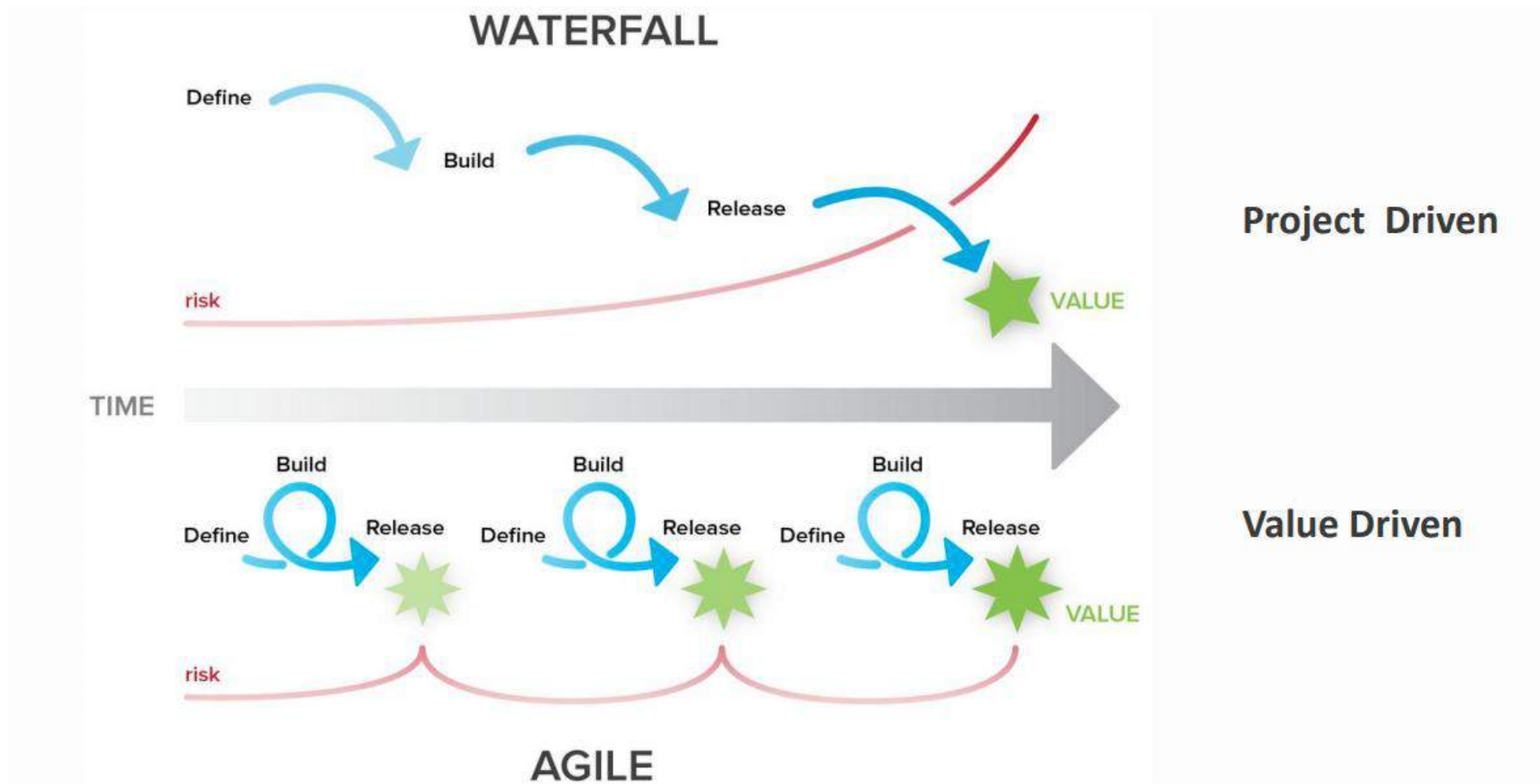
● 稳态，清晰具体的  可预测的结果

● 敏态，模糊方向性的  成果交付达成

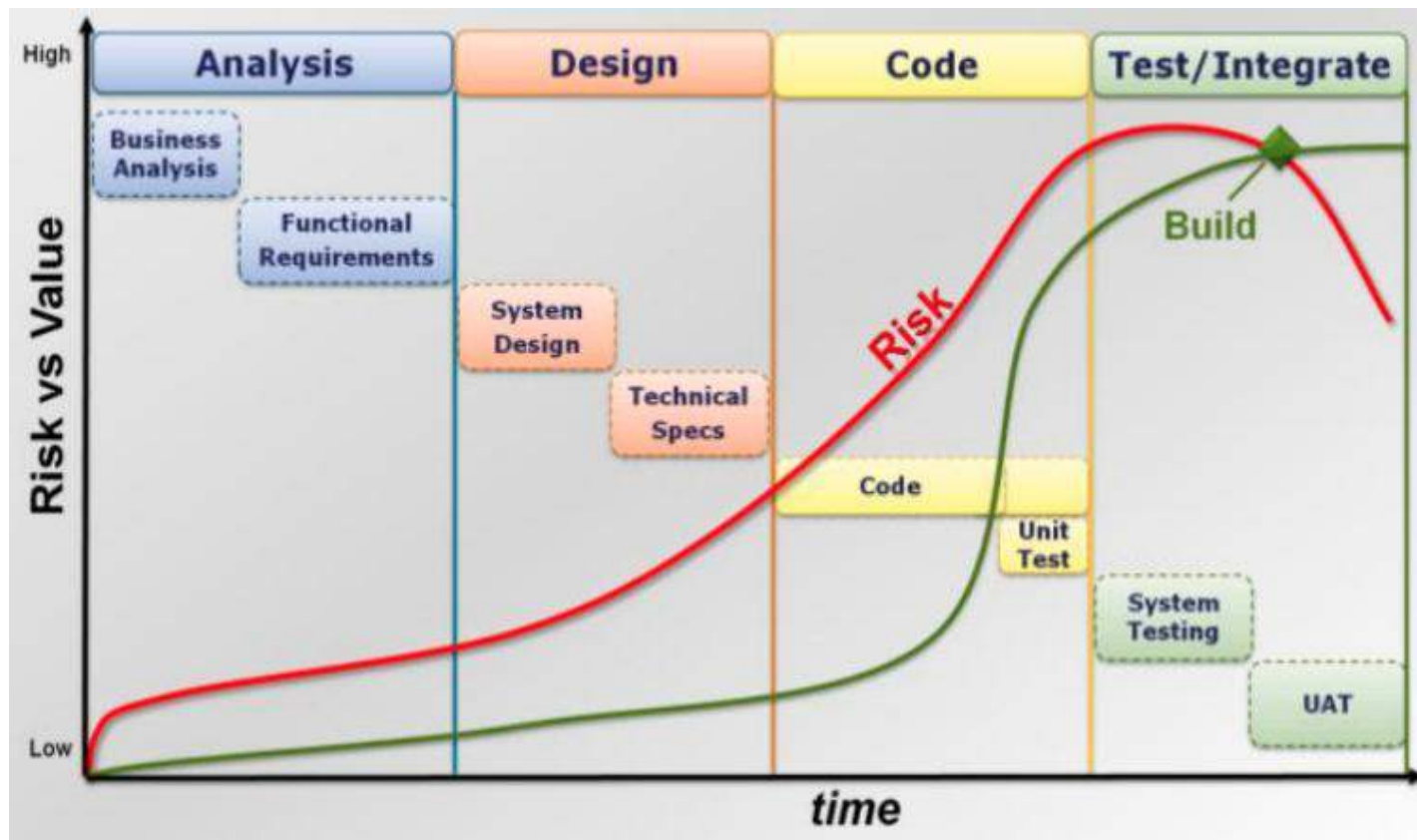
敏捷开发



瀑布模型和敏捷模型

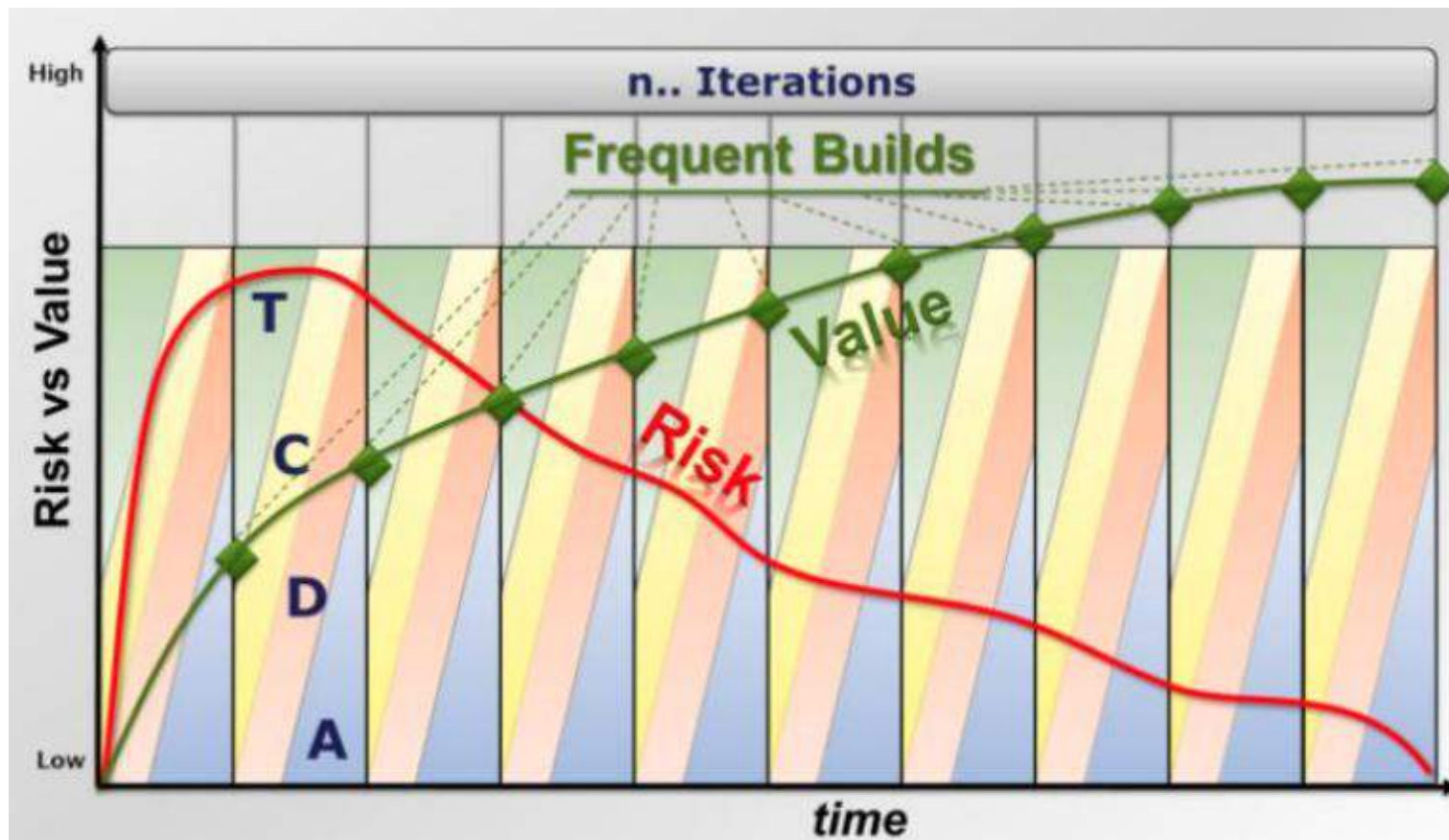


瀑布模型常见挑战



- 一直拖延的 Deadline
- 问题暴露晚，测试压力大
- 项目进度失控
- 功能无法按期交付
- 交付的不是客户预期的

敏捷模型常见挑战



- 需要客户参与和反馈
- 对团队成员技术要求高
- 对团队成员文化要求高
- 交付成本可能很高

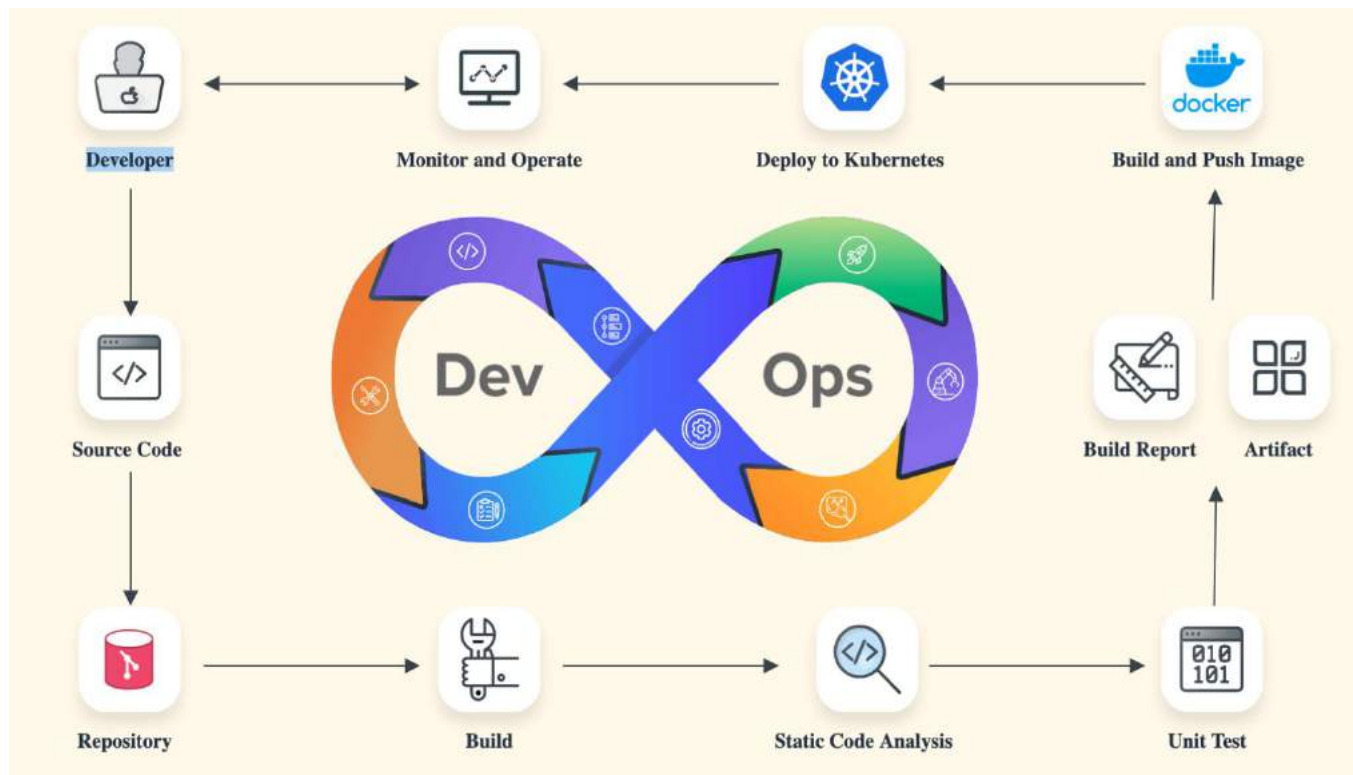
瀑布与敏捷模式对比

- 定义好范围，计划驱动，可预见性高
 - 瀑布式的过程
 - 严格控制“变更”
 - 成功 = 严格遵循项目计划和过程
 - 进度 = 交付件，如 需求规格说明书/设计文档/代码/测试案例/评审记录/报告报表
 - 命令与控制式的管理
- 务价值驱动，下一步行动来自当前认知
 - 迭代的过程
 - 拥抱变化
 - 成功 = 业务价值最大化
 - 进度 = 每迭代产出潜在可发布的产品增量
 - 自组织团队 + 服务型领导

场景讨论 - 怎么回复

开发经理走过来对你说：“我们需要开发一个新的 K8s 管理平台。它只需要能够管理 K8s，满足日常运维的功能，可以随时看到 K8s 集群的实时状况用来给老板汇报用。这个需要多长时间？”

DevOps 工具链



- 需求管理工具 Jira
- 知识管理工具 Confluence
- 版本控制系统 GitLab
- 持续集成工具 Jenkins
- 代码质量工具 SonarQube
- 构建工具 Maven/Gradle
- 制品管理 Artifactory/Harbor
- 配置中心 Apollo
- 测试工具 RF/Selenium/Jmeter

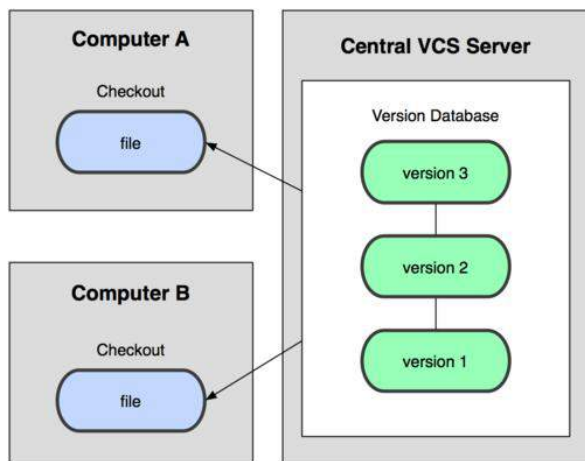
Git

Linus 在1991年创建了开源的 Linux，Linux 的代码是如何管理的呢？Linus 花了两周时间自己用 C 写了一个分布式版本控制系统，这就是 Git！

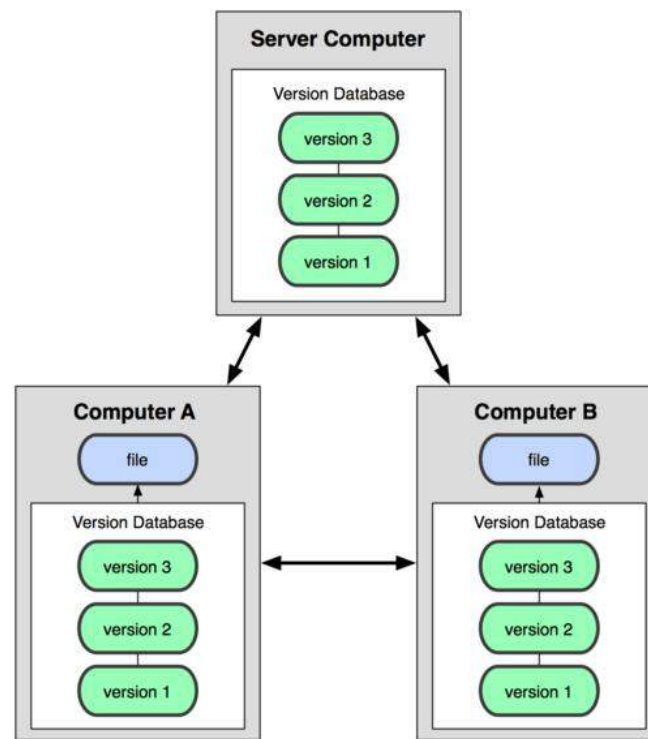
Git 是目前世界上最先进的分布式版本控制系统，那什么是版本控制系统？

- 能够记录文档的改动历史
- 能够回滚到任意一个修改时间点
- 能够协作编辑文件

集中式与分布式区别

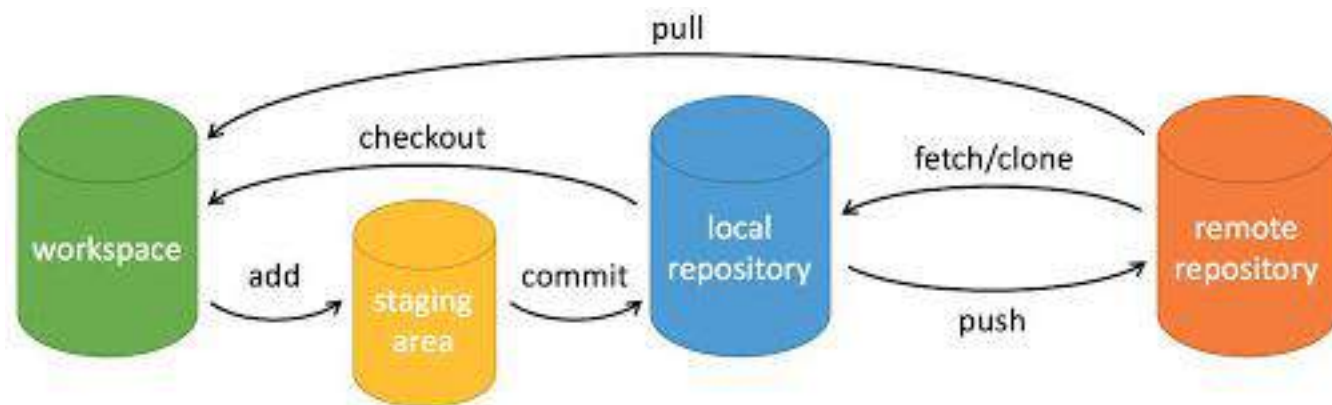


集中式: 客户端取出最新的文件进行修改, 服务端进行版本的管理



分布式: 客户端具有完整的统版本记录, 可离线工作

Git 的基本操作



git clone

git pull

git checkout

git branch

git add

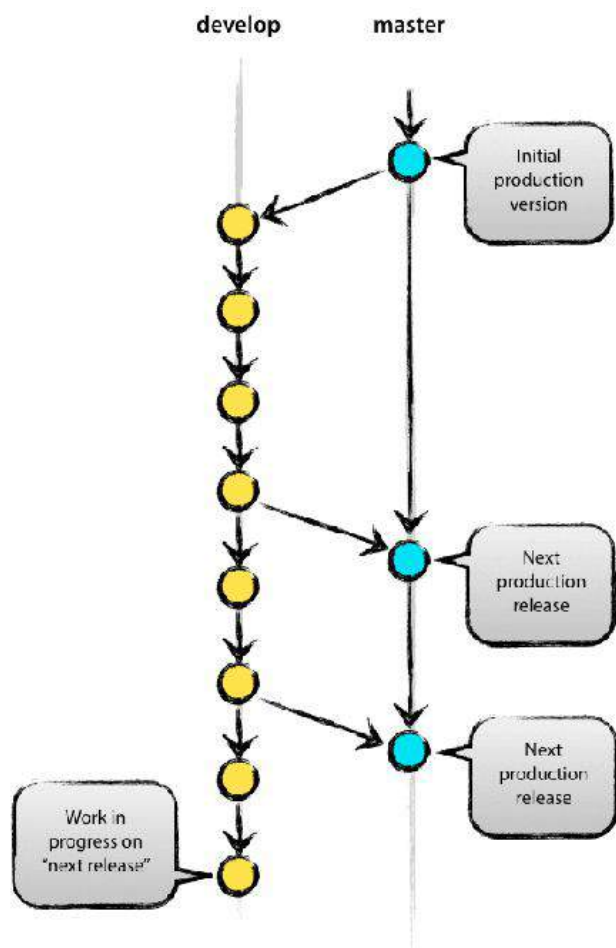
git commit

git push

随堂实验 01 - Git 的使用

- 注册 Gitee 账号
- 新建一个与自己账号同名的仓库
- 给仓库添加 GPL V3 协议
- 给仓库添加 Readme.md 文件
- 初始化提交一些项目文件
- 加入组织

分支模型 - Git flow



项目存在两个长期分支:

主分支 master

开发分支 develop

优点:

开发分清晰可控

缺点:

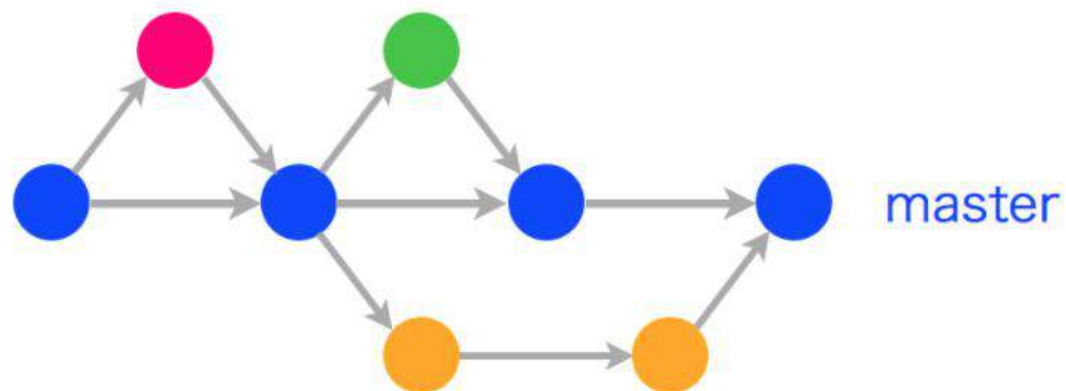
相对复杂，需要同时维护两个长期分支 develop

适用于:

发布周期比较长的项目

分支模型 - GitHub flow

GitHub flow



项目存在一个长期分支:

主分支 `master`

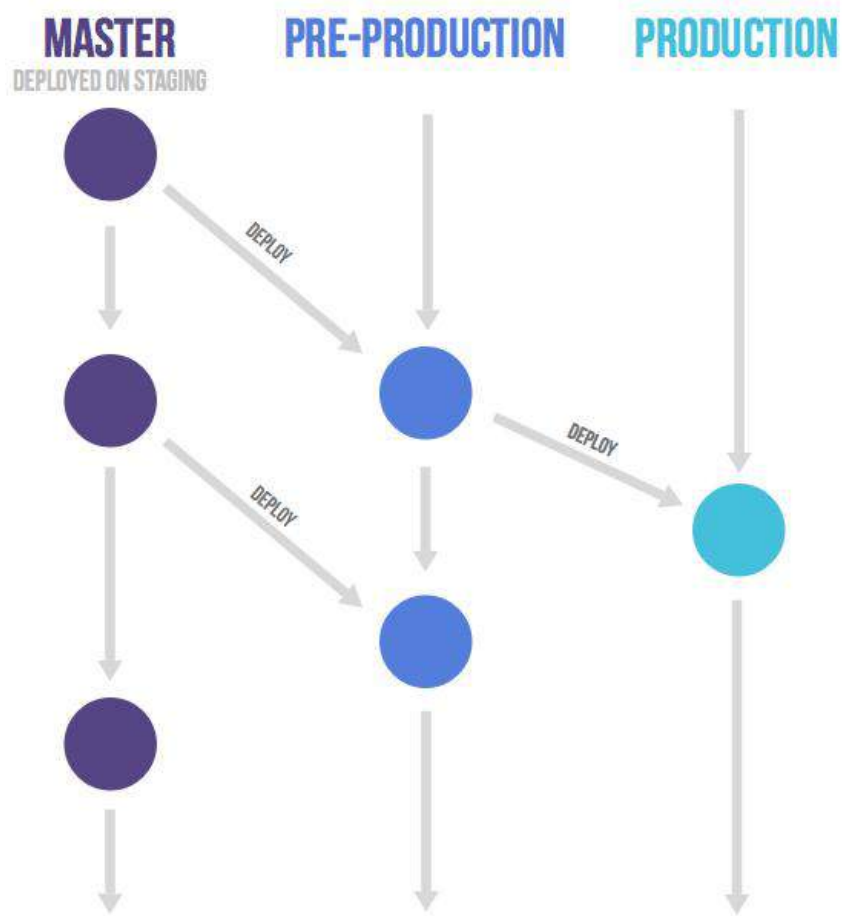
优点:

简单

适用于:

持续发布的产品

分支模型 - Gitlab flow



项目存在多个长期分支:

主分支master

下游分支

适用于:

Android

Chromium

具有上下游关系的项目

Jira

JIRA 是 Atlassian 公司出品的项目与事务跟踪工具。

被广泛应用于缺陷跟踪、客户服务、需求收集、流程审批、任务跟踪、项目跟踪和敏捷管理等工作领域，其配置灵活、功能全面、部署简单、扩展丰富。

快速部署:

```
docker run -p 8080:8080 cptactionhank/atlassian-jira:latest
```

Jira

Jira

Your work

Projects

Filters

Dashboards

People

Plans

Apps

Create

Q Search

Teams in Space

Classic software project

Scrum: Teams in S...

Board

Roadmap

Backlog

Active sprints

Reports

Issues

Components

Releases

Project pages

Add item

Project settings

Board

Q Quick Filters

TO DO 5

Engage Jupiter Express for outer solar system travel

SPACE TRAVEL PARTNERS

5

TIS-25

Create 90 day plans for all departments in the Mars Office

LOCAL MARS OFFICE

9

TIS-12

Engage Saturn's Rings Resort as a preferred provider

SPACE TRAVEL PARTNERS

3

TIS-17

Enable Speedy SpaceCraft as the preferred

IN PROGRESS 5

Requesting available flights is now taking > 5 seconds

SEESPACEEZ PLUS

3

TIS-8

Engage Saturn Shuttle Lines for group tours

SPACE TRAVEL PARTNERS

4

TIS-15

Establish a catering vendor to provide meal service

LOCAL MARS OFFICE

4

TIS-15

Engage Saturn Shuttle Lines for group tours

CODE REVIEW 2

Register with the Mars Ministry of Revenue

LOCAL MARS OFFICE

3

TIS-11

Draft network plan for Mars Office

LOCAL MARS OFFICE

3

TIS-15

DONE 8

Homepage footer uses an inline style - should use a class

LARGE TEAM SUPPORT

TIS-68

Engage JetShuttle SpaceWays for travel

SPACE TRAVEL PARTNERS

5

TIS-23

Engage Saturn Shuttle Lines for group tours

SPACE TRAVEL PARTNERS

TIS-15

Establish a catering vendor to provide meal service

Release

Issue 是什么

Issue 就是一个工作任务

- 准备去做的事
- 解决一个软件bug
- 一个客户需求
- 一个想法
- 一则招聘信息

Issue 包含任务相关的信息

- 谁生成的任务
- 谁负责此任务
- 时间
- 任务状态

Issue 的操作

- 创建

- 选择 Issue 类型
- 输入信息：概要、优先级、模块、版本、经办人、描述、附件

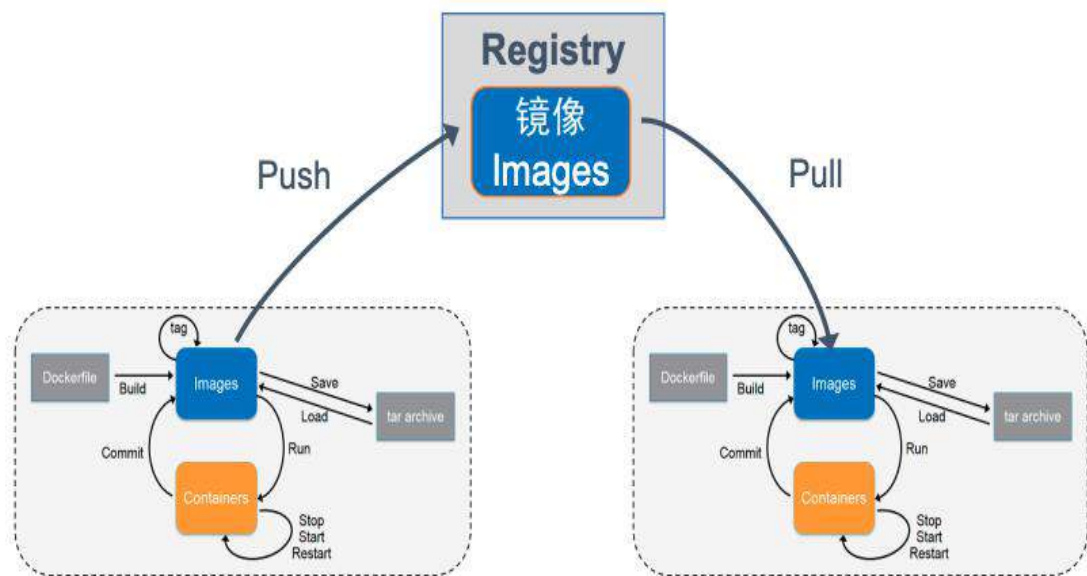
- 管理

- 添加附件
- 移动到其它项目、修改 Issue 类型
- 与其他 Issue 关联
- 创建子任务
- 记录工作日志

随堂实验 02 - issues 使用

1. 创建 Issue
2. 给 Issue 添加模块信息
3. 给 Issue 设置时间范围
4. 给 Issue 添加标签
5. 在搜索界面找到 Issue
6. 在 Issue 中 at 其他人

Registry



Registry

- 镜像存储仓库
- 分发镜像的媒介
- 访问控制
- 镜像管理

Harbor

开源企业级容器镜像仓库

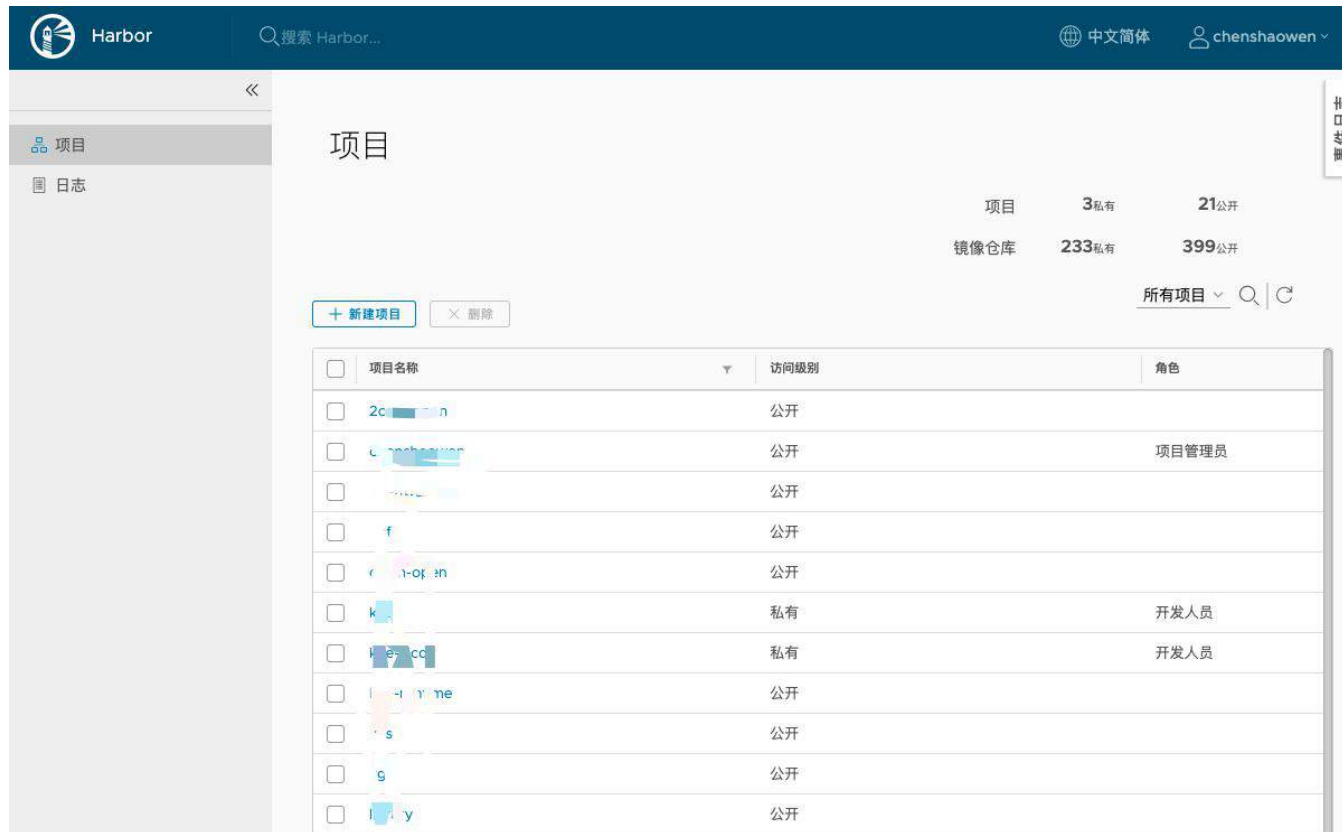
- 由 VMware 中国团队设计和开发
- 集成到多个企业级产品中
- Apache 2 使用许可
- <https://github.com/vmware/harbor/>



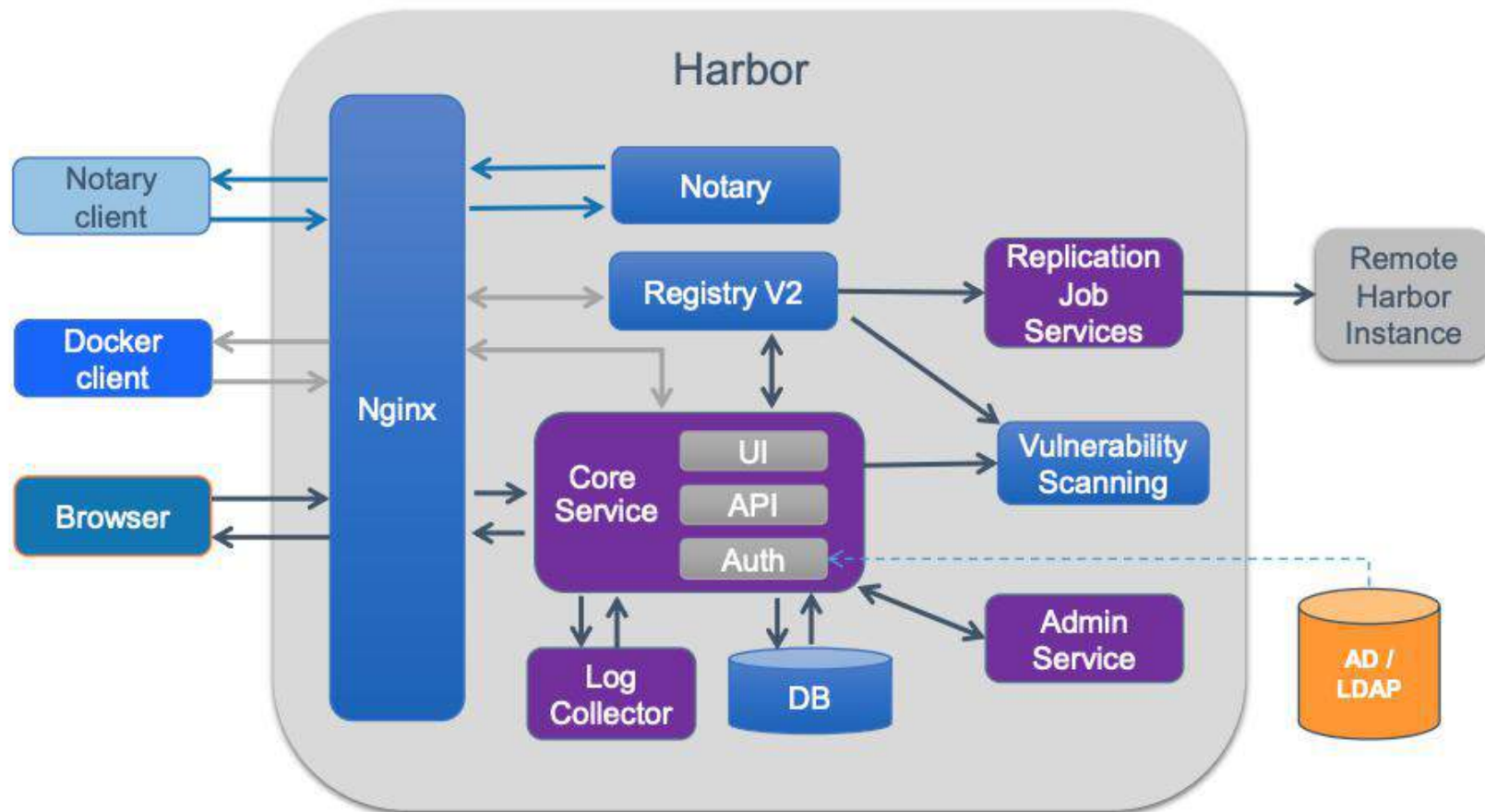
Harbor 主要功能

用户管理和访问控制

- 基于角色的访问控制 (RBAC)
- AD/LDAP 用户身份集成
- 镜像远程复制
- 镜像安全漏洞扫描
- 镜像来源公证 (content trust)
- 图形化管理界面
- 审计和日志
- Restful API



Harbor 的架构



Harbor 是生产环境中，非常常见的镜像仓库服务。

有时，还会结合 Dragonfly 等组件来使用。

问答

- DevOps 是什么?
- CI/CD 与 DevOps 的区别?
- 说出几个 DevOps 常用 工具?

Jenkins 持续集成

目录

- Jenkins 介绍
- Jenkins 架构
- 流水线
- 问答与实验

持续集成的作用

- 项目最后做模块集成的时候，发现很多接口不通
- 安装依赖包时，发现没有可用的软件包
- 搭建测试环境时，需要手工去解压包，修改配置
- 当团队成员，想要了解项目状态时，如何去呈现

持续集成就是用来解决以上问题，减少重复的步骤，降低项目的风险，任何时间、任何地点都可以生成可用的软件，增强项目的可见性。

Jenkins 介绍

- Jenkins 的前身是 Hudson，采用 Java 编写的持续集成开源工具。
- Hudson 由 Sun 公司在 2004 年启动，第一个版本于 2005 年在 java.net 发布。
- 2007 年开始 Hudson 在构建领域占据领先地位
- 2010 年，因为 Oracle 对 Sun 的收购带来了 Hudson 的所有权问题。主要的项目贡献者和 Oracle 之间，尽管达成了很多协议，但有个关键问题就是商标名称 “Hudson”。
- 甲骨文在 2010 年 12 月声明拥有该名称并申请商标的权利。因此，2011 年 1 月 11 日，有人要求投票将项目名称从 Hudson 改为 Jenkins。
- 2011 年 1 月 29 日，该建议得到社区投票的批准，创建了 Jenkins 项目。

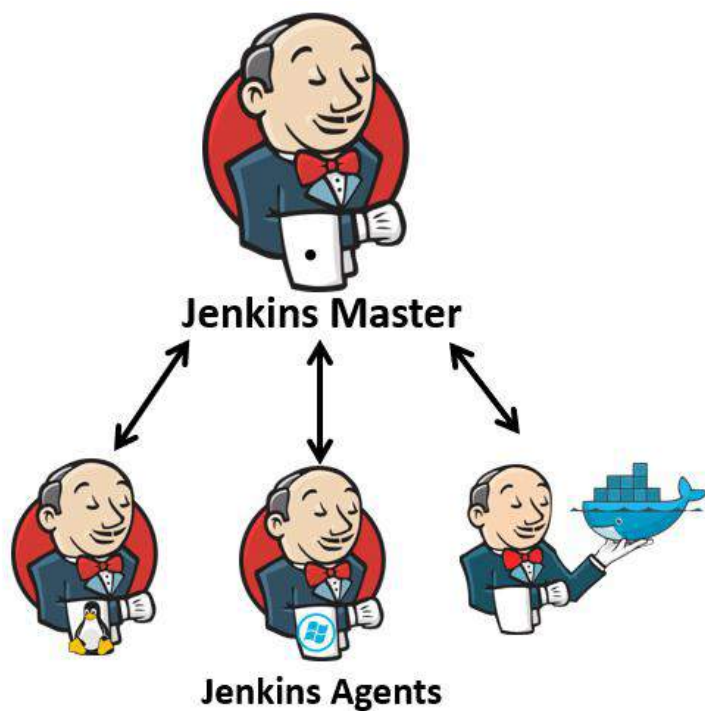
Jenkins 的功能特点

- 开源免费
- 多平台支持（windows/linux/macos）
- 主从分布式架构
- 提供web可视化配置管理页面
- 安装配置简单
- 插件资源丰富

Jenkins 应用场景

- 集成svn/git客户端实现源代码下载检出
- 集成maven/ant/gradle/npm等构建工具实现源码编译打包单元测试
- 集成sonarqube对源代码进行质量检查（坏味道、复杂度、新增bug等）
- 集成SaltStack/Ansible实现自动化部署发布
- 集成Jmeter/Soar/Kubernetes/.....
- 可以自定义插件或者脚本通过jenkins传参运行
- 可以说Jenkins比较灵活插件资源丰富，日常运维工作都可以自动。

Jenkins 架构



- Master 节点作为协调者，负责与执行节点通信并分派任务
- Agent节点任务执行者

随堂实验 01 - Jenkins 的安装

方式一：原生安装包

```
apt-get install jenkins
```

方式二：直接运行 jenkins.war

方式三：Docker 或者 Kubernetes 安装

```
docker run -p 8080:8080 --name myjenkins -v jenkins_home:/var/jenkins_home -  
v jenkins_downloads:/var/jenkins_home/downloads jenkins/jenkins:its
```

参考： <https://www.jenkins.io/zh/download/>

Jenkins Web 页面

Jenkins

🔍 查找

🔔 4

👤 admin

🚪 注销

Jenkins

新增任务

用户列表

构建历史

系统管理

Support

我的视图

打开 Blue Ocean

Labels Dashboard

Lockable Resources

新建视图

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

3 空闲

4 空闲

5 空闲

所有

+

S	W	名称 ↓	上次成功	上次失败	上次持续时间	收藏
		init-job	没有	2 月 20 天 - #1	:98 毫秒	
		test7wmp1	没有	无	无	

图标: 小 中 大

图谱

Atom feed 全部

Atom feed 失败

Atom feed 最新的构建

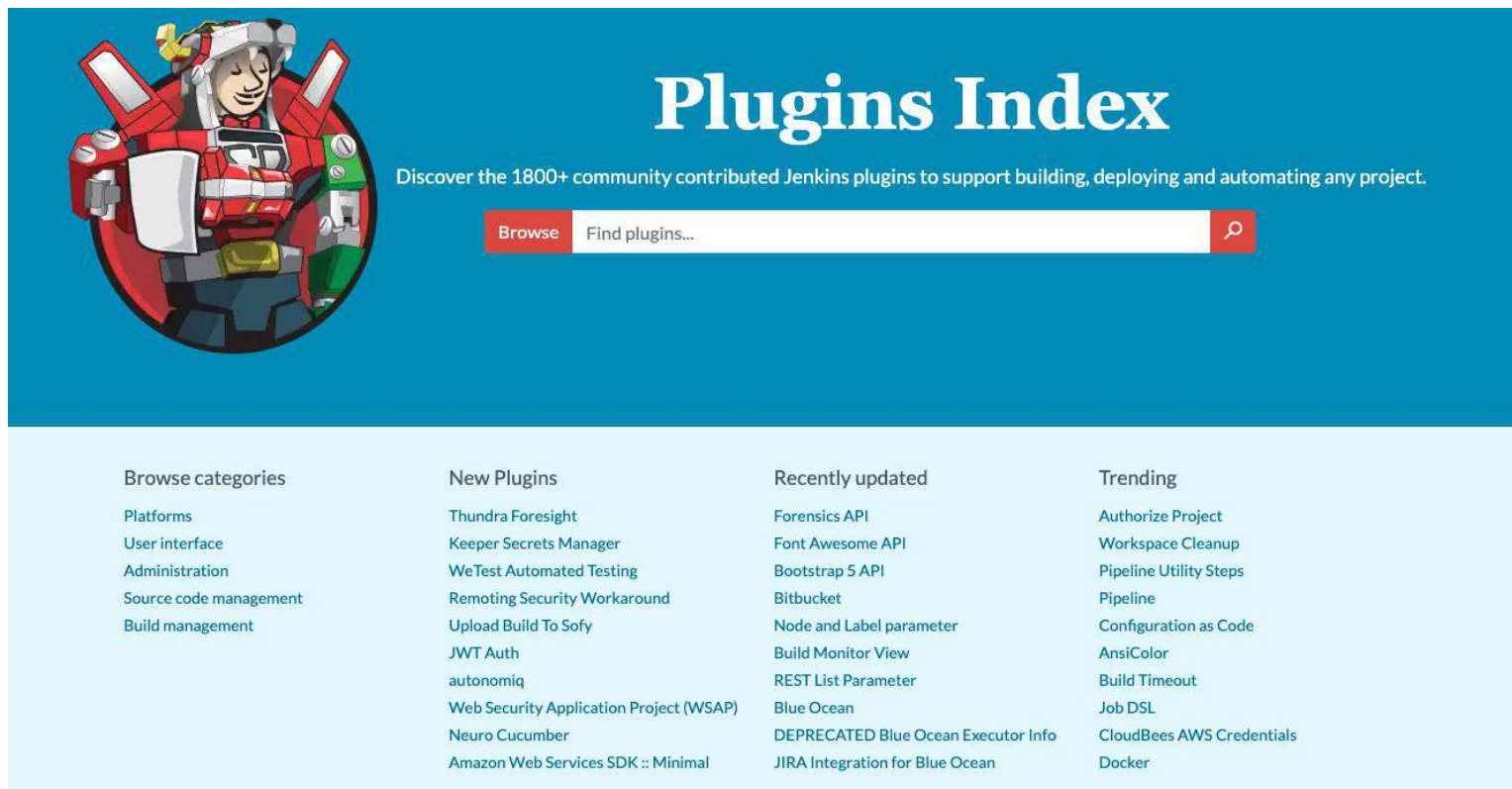
jenkins 中文社区

REST API | Jenkins 2.249.1

Jenkins 插件的安装

方式一：官网下载

<https://plugins.jenkins.io/> 使用 jenkins-plugin-cli 命令安装

The image shows the Jenkins Plugins Index page. At the top left is the Jenkins logo, a cartoon robot. To its right is the title "Plugins Index" in large white font. Below the title is a subtitle: "Discover the 1800+ community contributed Jenkins plugins to support building, deploying and automating any project." Below this is a search bar with a red "Browse" button and a magnifying glass icon. The page is divided into four columns: "Browse categories", "New Plugins", "Recently updated", and "Trending". Each column lists several plugins or categories.

Plugins Index

Discover the 1800+ community contributed Jenkins plugins to support building, deploying and automating any project.

Browse Find plugins...

Browse categories	New Plugins	Recently updated	Trending
Platforms	Thundra Foresight	Forensics API	Authorize Project
User interface	Keeper Secrets Manager	Font Awesome API	Workspace Cleanup
Administration	WeTest Automated Testing	Bootstrap 5 API	Pipeline Utility Steps
Source code management	Remoting Security Workaround	Bitbucket	Pipeline
Build management	Upload Build To Sofy	Node and Label parameter	Configuration as Code
	JWT Auth	Build Monitor View	AnsiColor
	autonomiq	REST List Parameter	Build Timeout
	Web Security Application Project (WSAP)	Blue Ocean	Job DSL
	Neuro Cucumber	DEPRECATED Blue Ocean Executor Info	CloudBees AWS Credentials
	Amazon Web Services SDK :: Minimal	JIRA Integration for Blue Ocean	Docker

Jenkins 插件的安装

方式二：管理页面安装

进入 Jenkins 后台操作

升级站点

URL

<https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json>

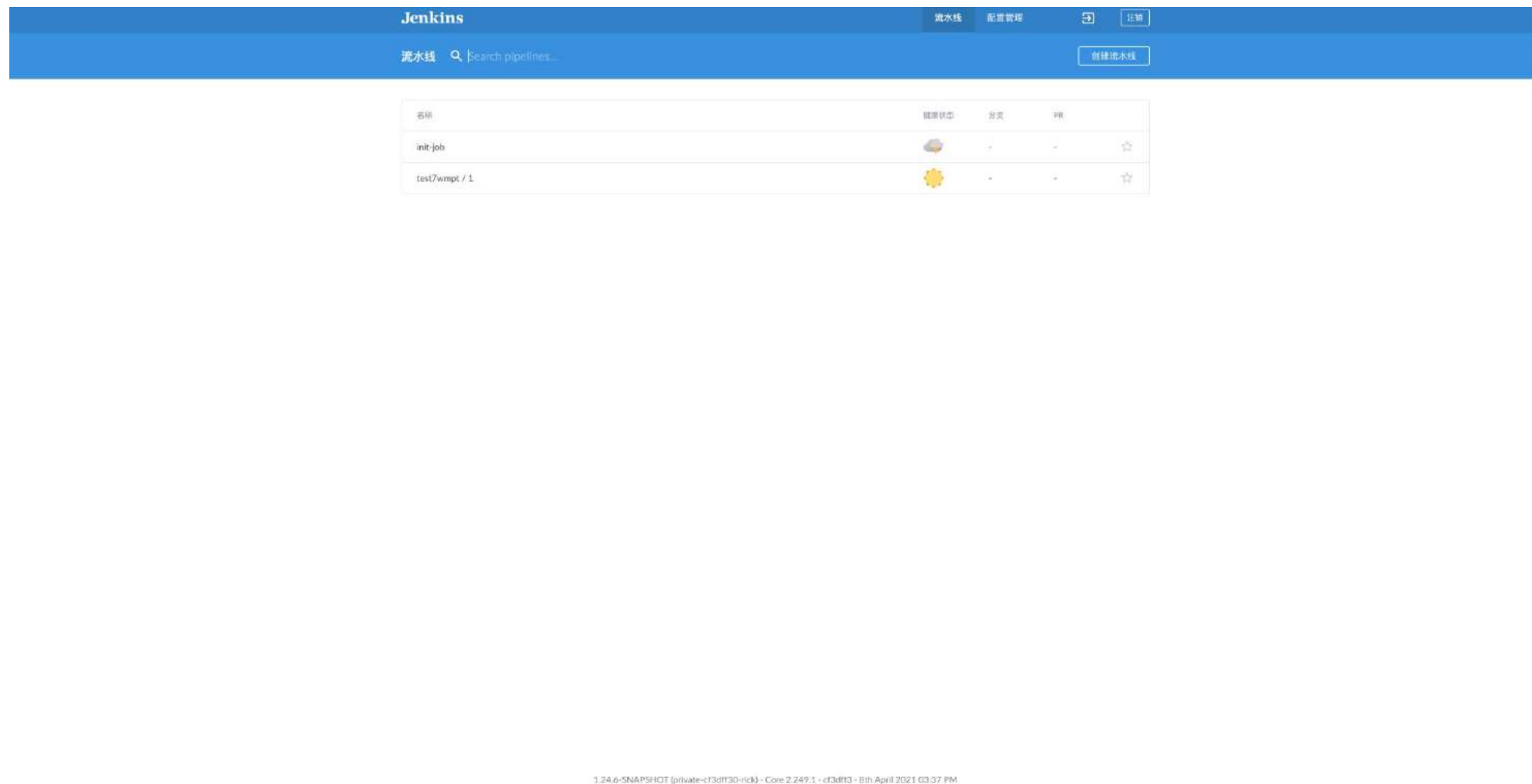
提交

Manage Jenkins (系统管理) -> Manage Plugins (插件管理) ->Advanced(高级)

->修改Update Site(升级站点):

<https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json>

也可以试试 Jenkins Blue Ocean 界面



Master 与 Agent 交互方式

- SSH

Master 能通过 SSH 直连 Agent

Agent 上启动了 SSHD 服务

- JNLP

Agent 能访问 Master

Agent 需要 JVM 环境运行

Master 需要放开 50000（默认值）端口用于 Agent 通信

- WebSocket

Agent 能访问 Master

Agent 需要 JVM 环境运行

Jenkins 通过标签选择

- 标签是什么

当 agent 数量变多时，如何知道哪些 agent 支持 JDK 8、哪些 agent 支持 Node.js 环境呢？

- 标签能做什么

就是给 agent 节点设置一个过滤的别名，在后续执行 job 的时候，可根据标签名字进行执行节点的分配。

Jenkins 页面中的标签

返回工作台

管理 Jenkins

新建节点

Configure Clouds

节点监控

构建队列

队列中没有构建任务

构建执行状态

build-agent-prod-11

1 空闲

build-agent-prod-12

1 空闲

build-agent-prod-22

1 空闲

build-agent-prod-11

2 kae-image-build-job #75190 (build)

名称 ↓

build-agent-prod-11

build-agent-prod-12

build-agent-prod-22

build-agent-prod-23

build-agent-prod-24

build-agent-prod-43

master

pri-beijing-cd-rysa

pri-zhuhai-ci

private-beta-cd

private-beta-ci

架构

Lin

Lin

Lin

Lin

Lin

Lin

Lin

Lin

Lin

Lin

获取到的数据

Jenkins

Dashboard 节点列表 build-agent-prod-11

返回列表

状态

删除节点

配置从节点

构建历史

负载统计

脚本命令行

日志

系统信息

断开连接

代理 build-agent-prod-11

Agent is connected.

=标签

kae-ci-build kae-ci-build-11

关联到build-agent-prod-11的项目

无

两种流水线使用

- 自由风格软件项目（FreeStyle Project）

将整个过程分为了源码管理、构建触发器、构建环境、构建、构建后操作等阶段，通过页面进行操作。

- 流水线项目（Pipeline Project）

通过 Groovy 语言编写 Jenkinsfile 定义流水线，更加灵活。

创建自由风格的 Job

输入一个任务名称

» 该字段不能为空，请输入一个合法的名称



构建一个自由风格的软件项目

这是Jenkins的主要功能.Jenkins将会结合任何SCM和任何构建系统来构建你的项目,甚至可以构建软件以外的系统.



流水线

精心地组织一个可以长期运行在多个节点上的任务。适用于构建流水线（更加正式地应当称为工作流），增加或者组织难以采用自由风格的任务类型。



构建一个多配置项目

适用于多配置项目,例如多环境测试,平台指定构建,等等。



文件夹

创建一个可以嵌套存储的容器。利用它可以进行分组。视图仅仅是一个过滤器，而文件夹则是一个独立的命名空间，因此你可以有多个相同名称的的内容，只要它们在不同的文件 夹里即可。



GitHub 组织

扫描一个 GitHub 组织（或者个人账户）的所有仓库来匹配已定义的标记。



多分支流水线

根据一个SCM仓库中检测到的分支创建一系列流水线。

如果你想根据一个已经存在的任务创建，可以使用这个选项

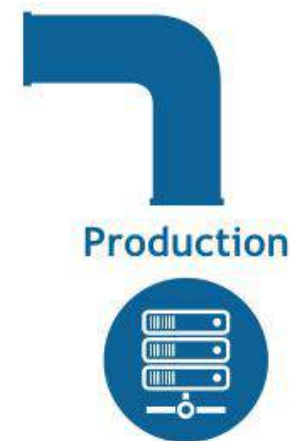
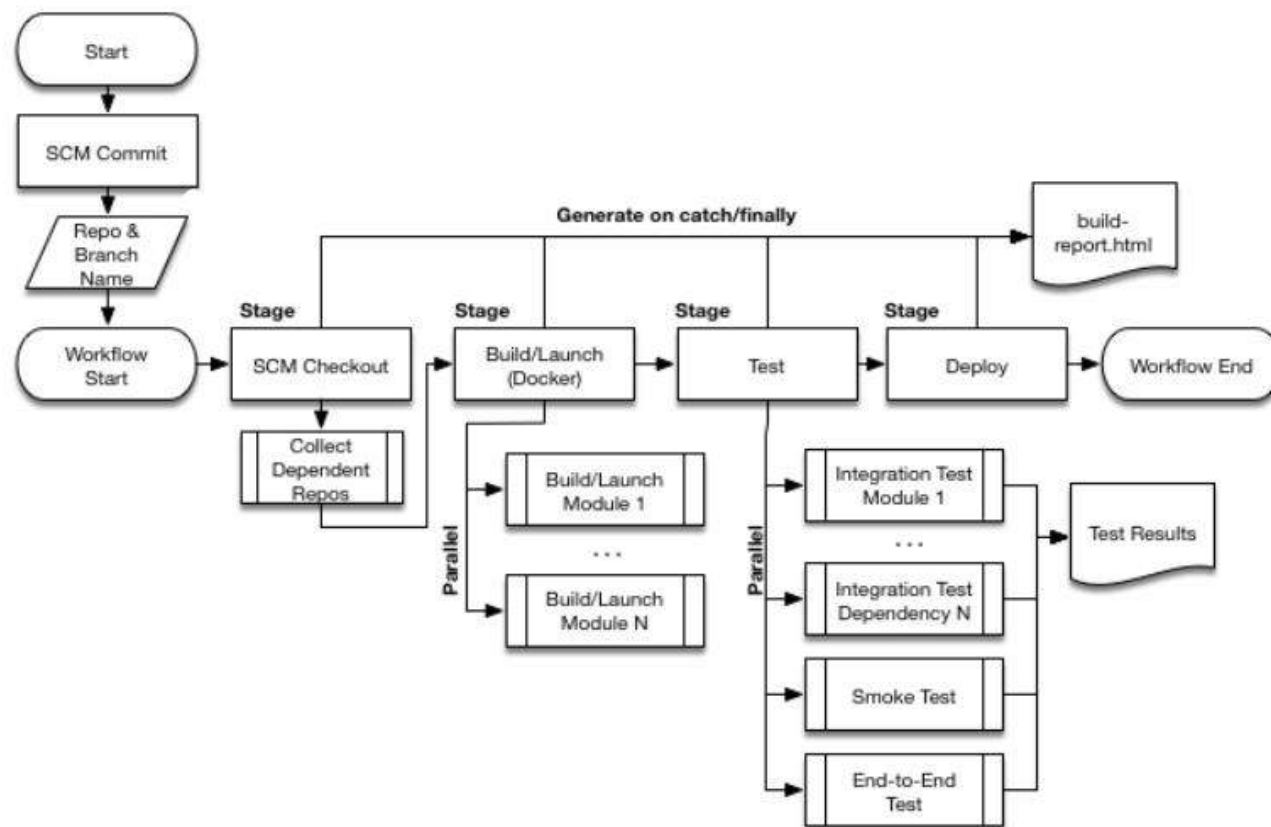
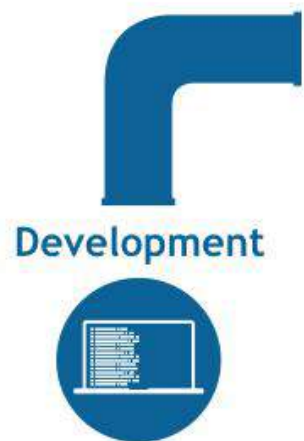


复制

输入自动完成

确定

Pipeline 风格的 job



Jenkins Pipeline

```
pipeline {  
    agent {  
        label 'java'  
    }  
    stages {  
        stage('Example Build') {  
            steps {  
                echo 'Maven Build Start'  
                sh 'mvn XXXXX'  
            }  
        }  
        stage('Example Deploy') {  
            steps {  
                echo 'Maven Build Start'  
                sh 'scp XXXX'  
            }  
        }  
    }  
}
```

问答 01

- Jenkins 有什么优缺点?
- Jenkins 有哪几种流水线?
- Jenkins 可以帮我们做哪些事?

随堂实验 02 - 尝试 Jenkins 的两种风格流水线

1，创建自由风格流水线并执行

2，使用 Jenkinsfile 文件定义流水线

可以执行的操作:

- Hello world
- git clone 代码
- 执行主机上的一些命令，docker build、uname -a