知识回顾：

1. SpringCloud
   1. Hystrix断路器、熔断器，写一个fallback方法，当程序失败时，不像传统程序，try/catch。提示极其不友好（失败）。它实现程序异常时，回调这个开发者自己写的方法。方法内返回默认值（预案）。防止分布式微服务结构出现雪崩。
   2. 微服务雪崩：很多服务组成一起完成一个业务，其中可能某一个服务出现异常，依赖它的这些服务都被阻塞。服务池溢出，所有服务都无法正常完成，就发生雪崩。Hystrix很好来解决雪崩问题。发现异常，快速尝试如果失败立即返回去调用fallback方法。给默认值返回用户。这些调用微服务，调用完成。清理现场，释放资源。
   3. Zuul API网关，授权、缓存、统一入口管理。配置映射。在application.yml Zuul路由，设置一个路径，路径作用就是分成不同模块。配置不同权限。
   4. NodeJS 基于chrome v8，号称目前为止解析js最快引擎。成为一个js解析引擎。类似tomcat。
   5. Sidecar本质，类似nginx配置一个连接而已。屏蔽技术壁垒。Web开发，url HTTP

今日任务：

1. Git 版本控制
2. STS eclipse-spring，创建一个真正springboot项目
3. springCloud-config配置中心

Git版本控制

Why?

历史：cvs、svn、git

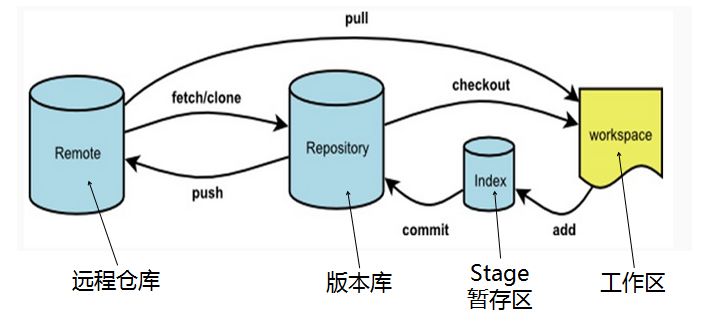
CVS，client-server结构，集中式部署，开发中文档管理（需求调研报告；需求说明书-每个字斟酌（产品经理）、架构师搭建架构、技术demo、界面原型开发、概要说明书（创建数据库模型PDM）、详细设计（伪代码）、开发、测试、运维、交付）file，bug比较多

SVN，client-server结构，集中式部署，开发中文档管理，tree，bug相对少，主流

很强权限管理

GIT，分布式部署，代码github（微软），tree。已经业界主流

没有学习SVN？藏拙，SVN不熟，但是我对GIT很熟悉，日常操作没问题。



\*Add 把工作区的代码增加到暂存区

\*Commit 把暂存区中的代码提交到本地仓库

+Push 把本地仓库的代码提交到远程仓库，代码共享

-Fetch/clone 克隆，刚到公司，要把远程仓库中的代码克隆到本地

+Pull 拉取，把远程仓库中的代码直接放在工作区

-Checkout 从本地仓库检出代码到工作区

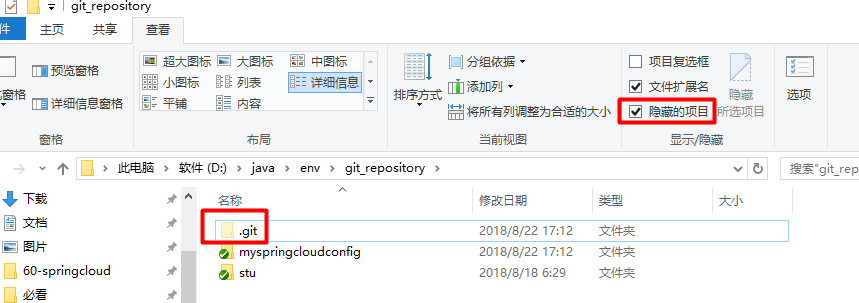
版本控制：（团队开发）

备份，如果文件被git管理，每次保存，会产生一个新的副本（只记录变化的内容）。V1.3+ time+作者（文件冲突）

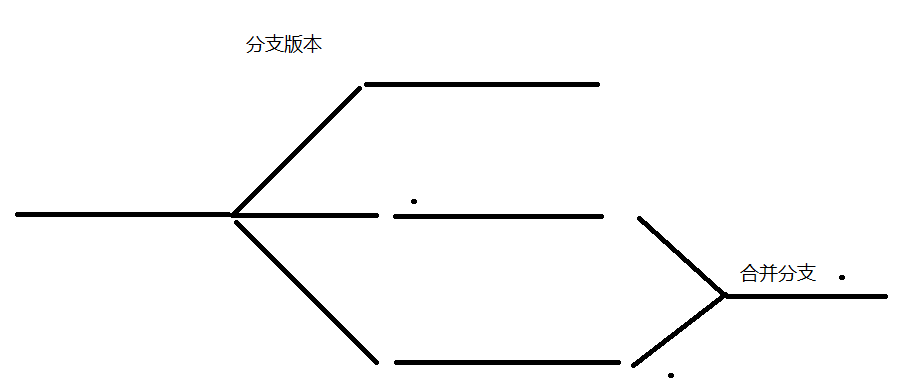
1. workspace工作区，你自己的代码在这个区域处理
2. Index索引，git独有，暂存区，存提交列表
3. Repository本地仓库，完成代码就存放在这里
4. Remote远程仓库，完成的代码，团队可以共同下载（检出）

结构和Maven非常相似

Github+git



重要概念：分支 branch



mysql被oracle公司收购。

1. mysql新版本收费
2. maridb 特点：基于之前开源版本的mysql
3. percona 京淘，基于之前开源版本的mysql

克隆：网络必须开启对github的访问，443端口

fatal: unable to access 'https://github.com/nutony/1805.git/': Unknown SSL protocol error in connection to github.com:443

致命：错误

Jdbc.properties上传？

用户名密码明码！root 123456 （Ysyhl9t 疑似银河落九天）

数据库允许外网直接访问！3306改掉

小乌龟客户端git，图标

1） 新文件，未受git管理

2） 代表文件属于暂存区，受git管理

3） 代表文件上传到远程仓库，受git管理

小乌龟：

add/commit/push/pull

配置中心SpringCloud ConfigServer

完成业务非常简单，之前配置信息存放本地文本文件中，属性文件，xml配置文件中。

，集群时手工修改每个配置文件，复制配置文件到其他节点上。不是动态。需要停止服务。传统加载方式都是在框架启动时加载配置项，一旦系统启动，就不会再去读配置文件。

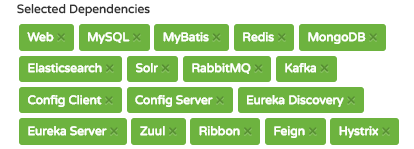
解决分布式结构

1. 分布式集群系统统一到一个配置中心去读取配置项
2. 动态刷新（并不是完全需要动态刷新）

创建SpringBoot应用程序，STS

Spring基于eclipse，插件，对spring整体应用程序支持更好，springboot直接支持

元数据metadata，描述项目的底层数据信息





application.properties和bootstra.properties有什么区别？

运行程序是有加载顺序。

注解@Value比application.properties执行早

解决：在@Value之前处理git事宜

Bootstrap.properties在application.properties执行前执行的。

动态配置刷新

1. SpringCloud增加@RefreshScope方式，手动刷新
2. SpringCloud Bus 服务总线（SOA ESB）MQ(RabbitMQ)

小结：

1. Git代码的版本控制，git分布式（可以暂时没有网络）svn、cvs集中式（必须连接到服务器才能工作）。Git不能完全替代SVN，Git管理是代码（公开），SVN管理项目文件（严格权限）
2. SpringCloud配置中心Config-Server。配置分布式化，配置动态。
3. 结构：config-client（消费者）启动时包扫描@Value注解，key=jtver，找spring框架，Spring在config-server启动时，就会加载所有的属性配置文件，就去config-server配置git仓库，把仓库中对应属性文件进行加载config-client(application)-dev(profile).properties文件。读取内容，hashmap（kv）。根据jtver到hashmap去找，获取到value值。把这个value设置，最终把这个获取到的值设置给controller的成员变量。在controller中就可以调用。