# 第一节课

# java的阶段

- Java基础
- Java SE阶段
- Java EE阶段 网络版
- Java企业级开发 Spring + SpringMVC + Mybatis（通过SSM框架架构实现了快速构建项目架构目的）

微服务分布式：减轻服务器负担。将一个服务器的职责分散出去。
熟练应用：javaSE，javaEE，Maven，Git，Mybatis，Spring，SpringMVC，日志框架等技术。
Linux操作以及Docker容器使用，Nginx，Redis，ElasticSearch中间件应用SpringBoot，RabbitMQ,SpringCloud微服务架构Zookeeper分布式解决方案。
架构：
1. 开发环境：安装很多软件，导致操作系统不安全，稳定性降低。
2. 生产环境：在生产中，操作系统不会采用win/MAc相对不安全，生产环境面向全体用户的，一般会采用专业的操作系统。
web1.0和web2.0
web1.0：带宽不足，这时项目内容少，用户量不多，甚至不需要对外开方，对安全性和稳定性要求不高，单体架构足以应付。

web2.0：实现ADSL（拨号上网）宽带提速，最高达8M，用户不断增加一些门户网站活跃，项目考虑安全性。基于1.0的单体架构，无法满足2.0对项目的需求。需要在单体架构上搭建集群。

[web2.0时期的垂直架构]
在搭建集群之后，可以提升项目稳定性，并且可以承受大量用户访问。
搭建集群后发生的问题
1.用户请求到底发送到那台服务器上，如何保证平均分发服务器，从而缓解用户量增加的压力
2.编写项目时，如果用户登录成功，将用户的表示写道session域中，数据共享的问题。
3.当数据量大时，直接查询数度很慢，提升查询效率
4.针对数据库SQL语句
Nginx 解决用户请求平均分发
Redis 解决数据共享并实现缓存
ElasticSearch 解决搜索数据的功能

比如项目中包含了三个模块 **用户模块，商品模块，订单模块**
商品模块压力过大，一般最直接有效的方式就时搭建集群，在单体架构的集群上搭建，效果相对比较差，随着项目的不断更新，项目中的功能越来越多，最严重可能导致项目无法启动，**关于单体架构**

**中，完美的体现了 低内聚 高耦合**

分布式架构

随着项目的不断迭代，老功能和信工之间需要相互交互，服务器和服务器之间需要通讯，项目一般分为三层，Controller,Service,Dao 导致程序变慢的重灾区 一般时service和Dao，在搭建集群时，确实针对三层都搭建集群，效果不时很好

**架构从垂直架构演变到了分布式架构**

分布式架构落地技术

国内通讯方式有两种

Dobbo RPC

SpringCloud HTTP

分布式架构常见问题

使用分布式架构之后，服务之间的通讯都是同步的

在一些不是核心业务的功能上 咱们希望实现异步通信，为类实现服务之间异步通讯 需要学习MQ(消息队列)

服务之间通讯地址的维护

由于服务越来越来，每个服务的访问地址都是一样的

协议://地址:端口号

由于模块繁多，并且模块搭建的集群数量增加，会到导致其他模块需要维护各种ip地址等信息，导致项目的维护性极低，耦合性变高，并且也无法实现负载均衡的功能 需要使用一个技术来决绝当前的问题：

Eureka注册中心 帮我们来管理服务信息

Robbin 可以帮我们实现服务之间的负载均衡:

服务降级

在上述的架构中，如果说订单模块出了问题，只要涉及到订单模块的功能，全部都无法使用，可能会导致，服务器提供的线程池耗尽，给用户友好的提示信息都无法做到，为了解决上述问题，使用Hystrix处理，Hystrix提供了线程池隔离的方式，避免服务器线程耗尽，在一个服务无法使用时，可以提供熔断器的方式来解决

Eureka,Robbin,Hystrix都是SpringCloud中的组件

海量数据

海量数据会导致数据库无法存储全部内容，即便数据库可以存储海量数据，在查询数据时，数据库的响应会及其缓慢，在用户高并发的情况下，数据库也无法承受住，为了解决上述的问题，可以基础MyCat实现数据库分库分表

微服务架构

虽然已经将每个模块独立的做开发，比如商品模块，压力最大的是商品查询，在单独模块中 再次拆分项目可以称之为微服务

模块过多，运维成本增加
为了解决模块过多 ， 运维成本增加的问题。采用Docker容器华技术来帮我们管理
JavaWeb 开发
CS架构
Client客户端/Service服务器端
自己编写客户端代码
自己定义客户端与服务器端的通讯协议
自己编写服务端代码
BS架构
Brows浏览器/Service 服务器端
浏览器(不用程序员自己写 有现成的)
通讯协议(浏览器与web服务器的通信协议为http)
服务器端(代码编写)
Servlet
sun 公司制定的扩展 web服务 器功能的 组件 规范
web服务基础功能 就是被浏览器访问
网站中的登录功能不是web服务器的默认功能，是程序员为web服务器添加的额外功能
step1 安装web服务器 tomcat
安装tomcat之前请先检查 本机jdk环境变量
win+r 进入cmd 虚拟Dos窗口
输入 java -version 回车 查看当前jdk版本
输入 javac 回车 查看javac的编译命令行
step2 如何实现Servlet的开发步骤
a,创建一个.java文件 让该类型继承HttpServlet 抽象类
b,对上一步编写完成的.java文件进行编译
javac -cp servlet-api.jar -d . HelloServlet.java
c,打包 即创建一个具有如下结构的文件夹
appname
WEB-INF
classes(放.class文件)
lib(放.jar文件的 可以有也可以没有)
web.xml(部署描述文件)
d,部署 将第三步打包好的文件 复制粘贴到tomcat 中一个叫webapps的文件夹中
e,启动tocmat服务器 启动成功后 打开浏览器
127.0.0.1:8080/appname/hello

```
1
2  import javax.servlet.*;
3  import javax.servlet.http.*;
4  import java.io.*;
5  public class HelloServlet extends HttpServlet{
6   public void service(HttpServletRequest request,HttpServletResponse response)throws
   ServletException,IOException{
7                   response.setContentType("text/html");
8                   PrintWriter out=response.getWriter();
9                   out.println("<h1>Hello Kitty</h1>");
10                  out.close();
11   }
12 }
```