

Technologies for E- Commerce

CAN302

**Department of Communications and Networking
Xi'an Jiaotong-Liverpool University (XJTLU)**

Week6 – Load and concurrence



Every request means...

- Network
- CPU
- Memory
- Disk I/O



- Concurrent user, Response Time, Throughput
- QPS (Query Per Second), TPS (Transaction Per Second)

The huge requests @ 12306



- 2020 Spring PV: 149,000,000,000

Distribute: Command many as ONE!



The same idea for Tesla and SpaceX

Powerful server and many servers

H3C UniServer R4900 G5 Server



Industrial-leading
performance improves data
center productivity

- Support up to 80 cores and 12TB memory
- 16 x Intel® Optane™ DC Persistent Memory Module(PMem 200)
- AI-Optimized GPU
- XG310 GPU for cloud gaming and media streaming

Highlights: High Performance High Reliability, High Scalability

New generation H3C UniServer R4900 G5 provide outstanding scalable capacity supporting up to 28 NVMe drives to enhance configuration flexibility for modern data centers

H3C UniServer R4900 G5 server is a H3C self-developed mainstream 2U rack server. R4900 G5 utilizes the most recent 3rd Gen Intel® Xeon® Scalable processors and 8 channel DDR4 memory with 3200MT/s speed to strongly lift the bandwidth up to 60% compared with previous platform.

With 14 x PCIe3.0 I/O slots and 2 xOCP 3.0 to reach excellent IO scalability.

Maximum 96% power efficiency and 5 ~ 45°C operating temperature provide users a TCO returns in a greener data center.

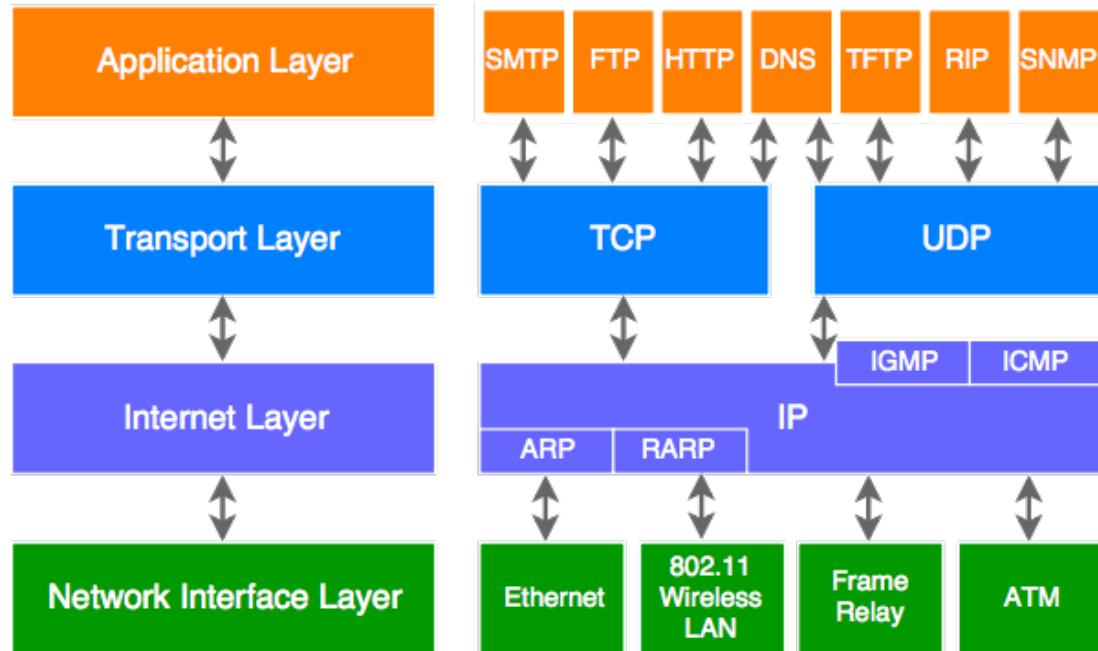


- Giants like Google, Facebook, Microsoft, Tencent … have more than million servers on-line.

<https://www.cnbeta.com/articles/tech/245359.htm>

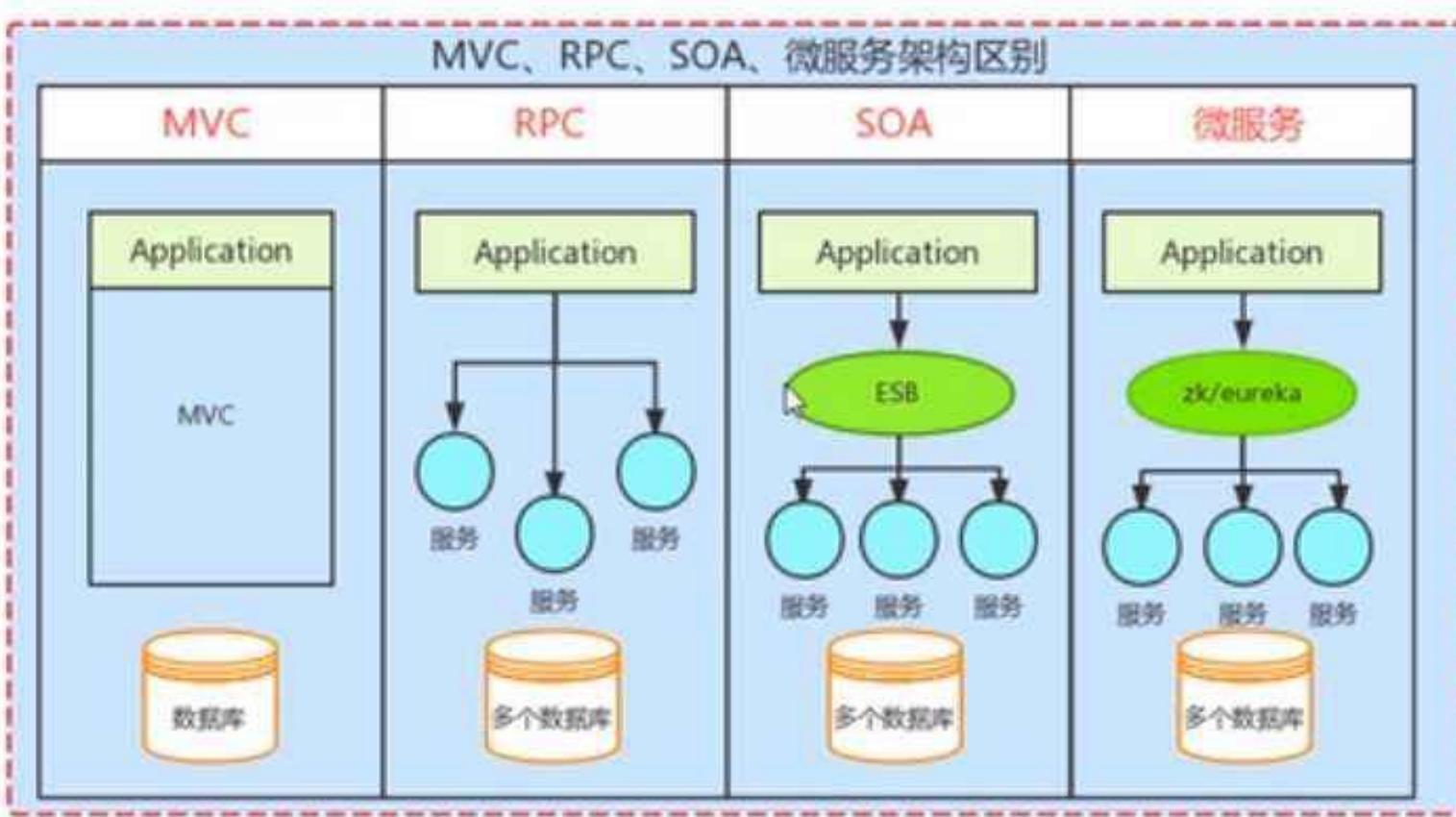
<https://baijiahao.baidu.com/s?id=1643175084761511155>

Load balance



- Re-direct the request
- Can be done in layer4, like LVS
- Can be done in layer7, like nginx

MVC to Microservice



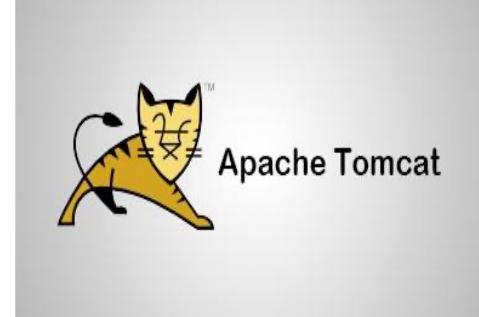
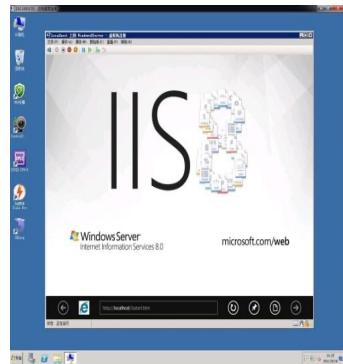
- The not change thing is to manage the spitted modules.
- PRC (Remote Procedure Call) combines multi-server as one.
- ESB manage more details on each service.
- Micro-service is better organized.

<http://m.mamicode.com/info-detail-2523756.html>

<https://blog.csdn.net/chushoufengli/article/details/104168347>

<https://www.zhihu.com/question/309621272/answer/577800728>

Web servers' performance are very different

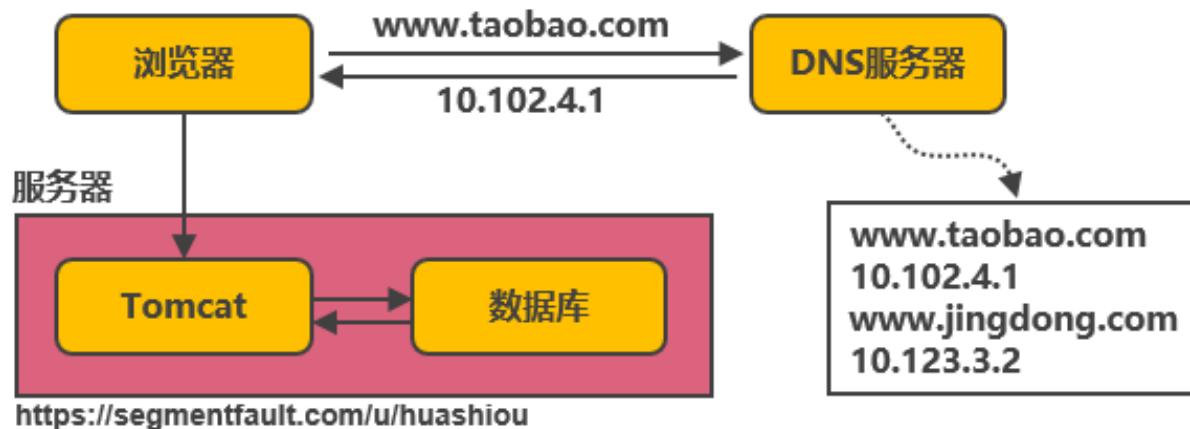


Typical concurrence:

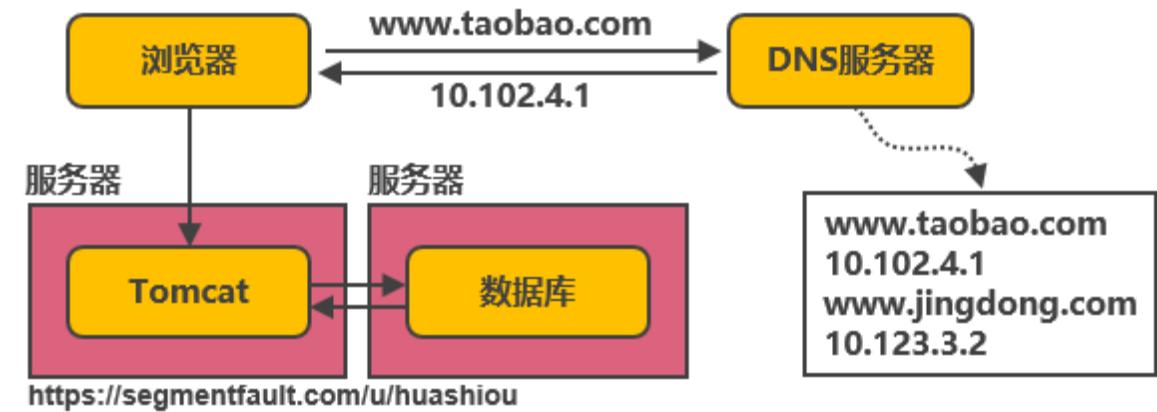
- Tomcat 300
- Apache 3000
- Nginx 30000

<https://www.zhihu.com/question/19571087>
<https://www.jianshu.com/p/5198359b59f7>

Stage 1



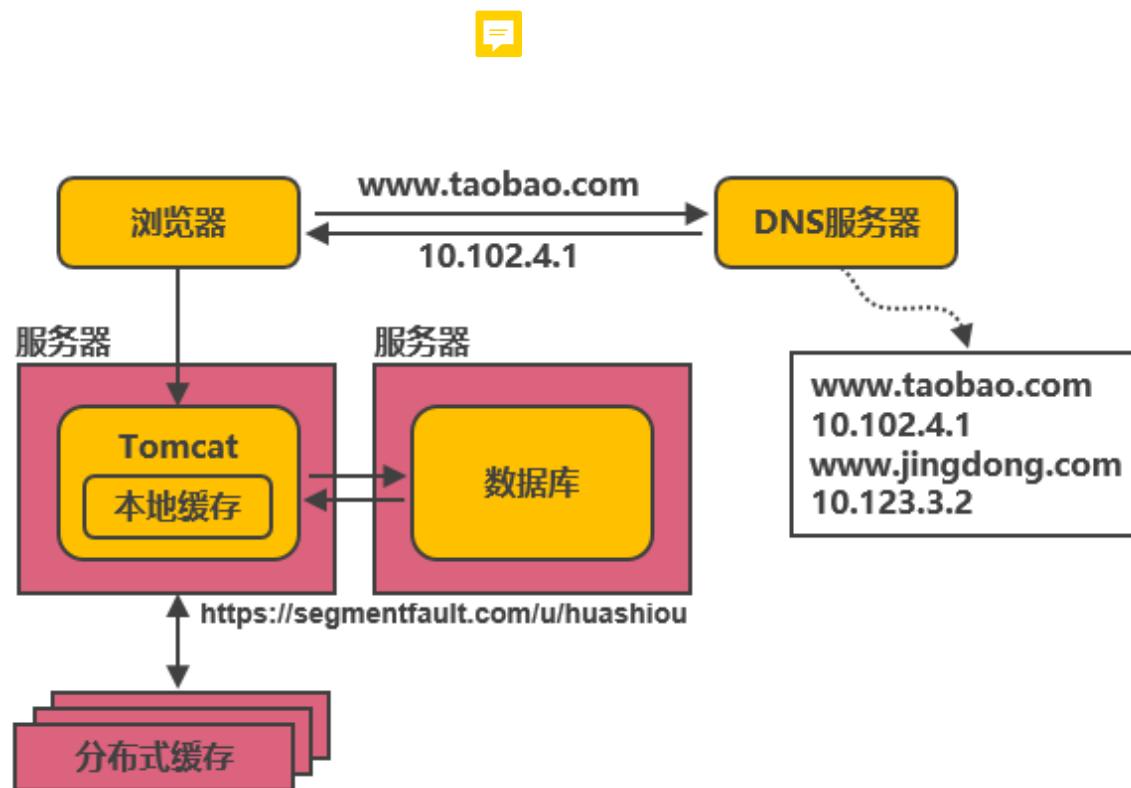
Let's start here.
May 200 con-current users.



Separate the web server and database

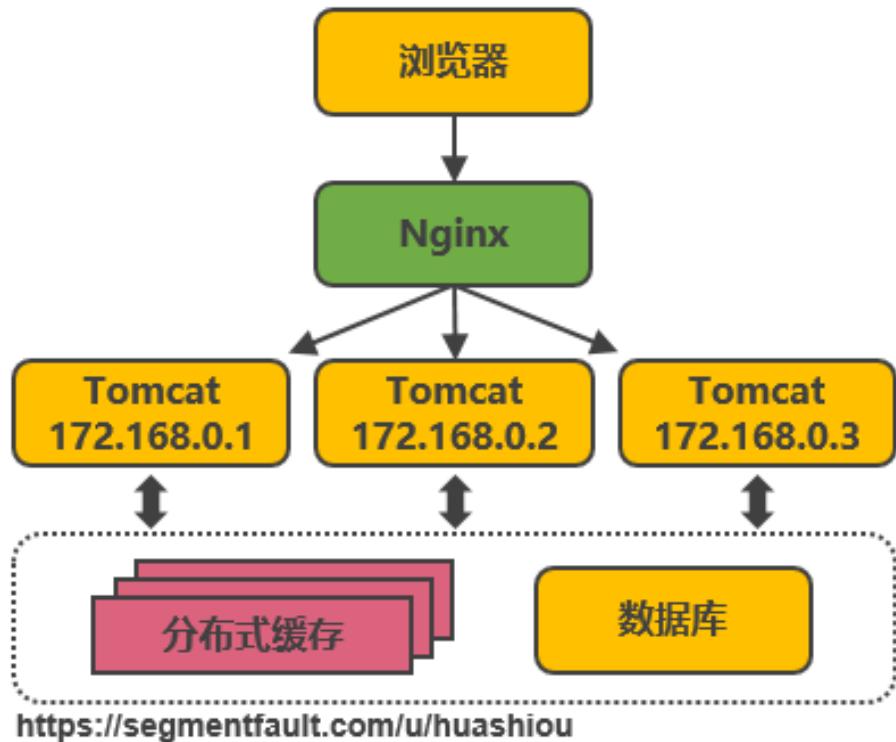
<https://segmentfault.com/a/1190000018626163>

Stage 2



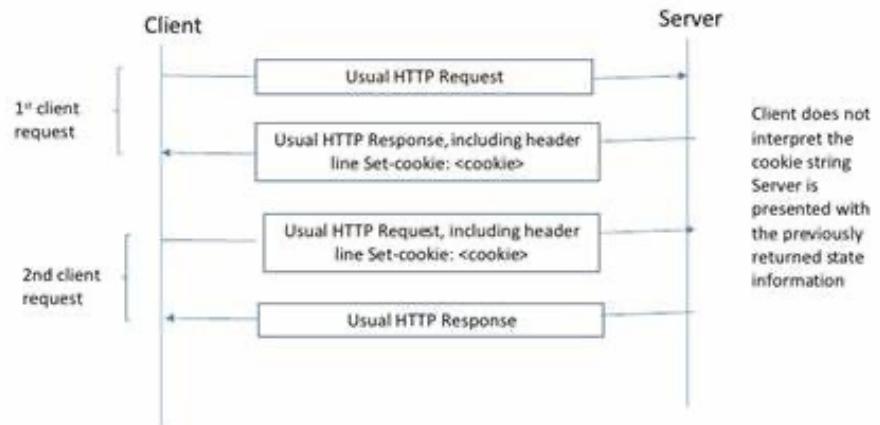
Using cache to reduce database queries

Stage 3



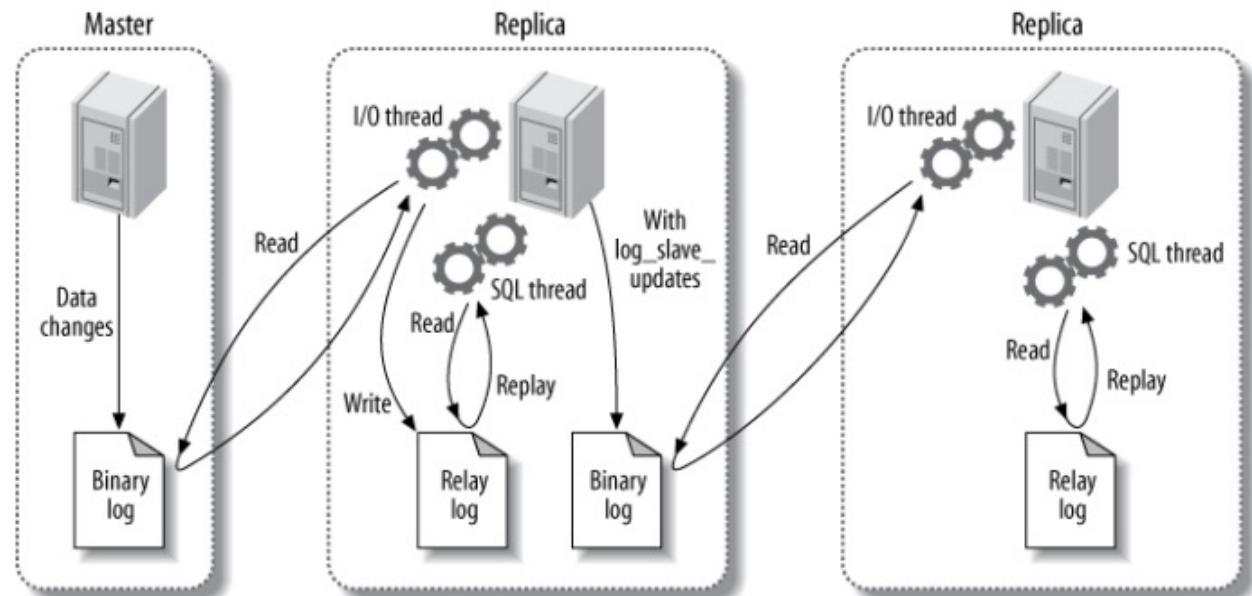
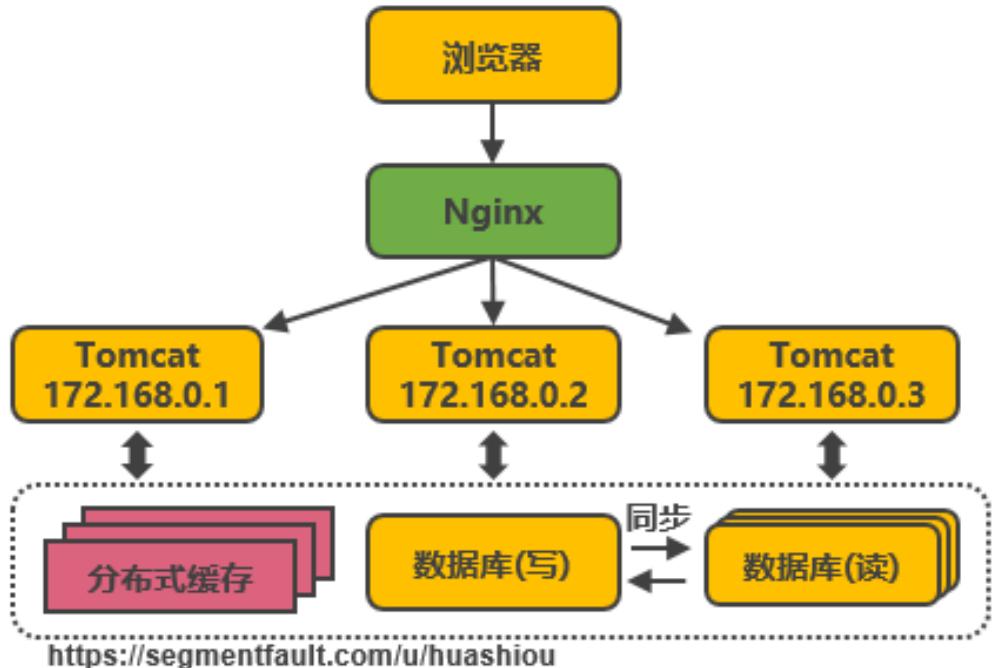
Cookies

- HTTP is stateless protocol
- Cookies manage state maintenance by shifting the burden to client
- Cookies are transmitted in clear text (security issue)



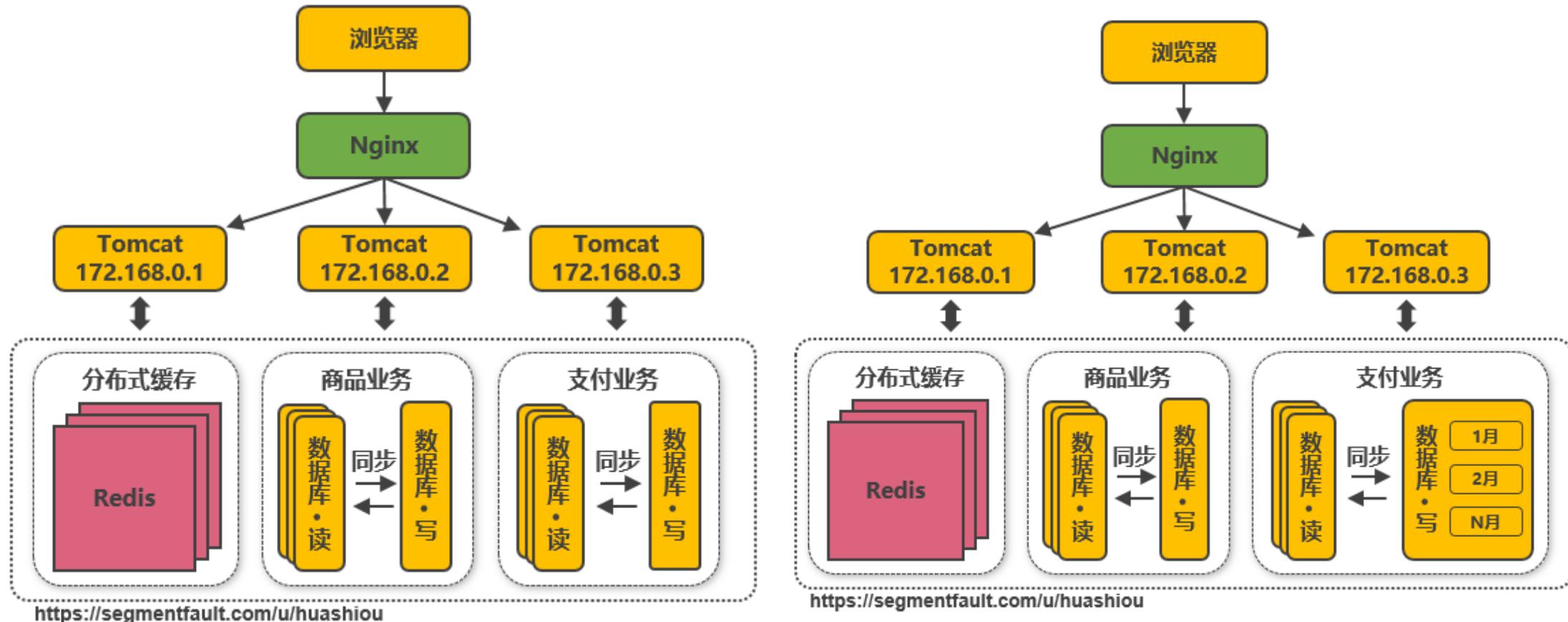
- **Using web proxy to have many web servers work together.**
- **Http is stateless, a web need cookie/session pair to keep it's status.**
- **Web servers need to share sessions.**

Stage 4



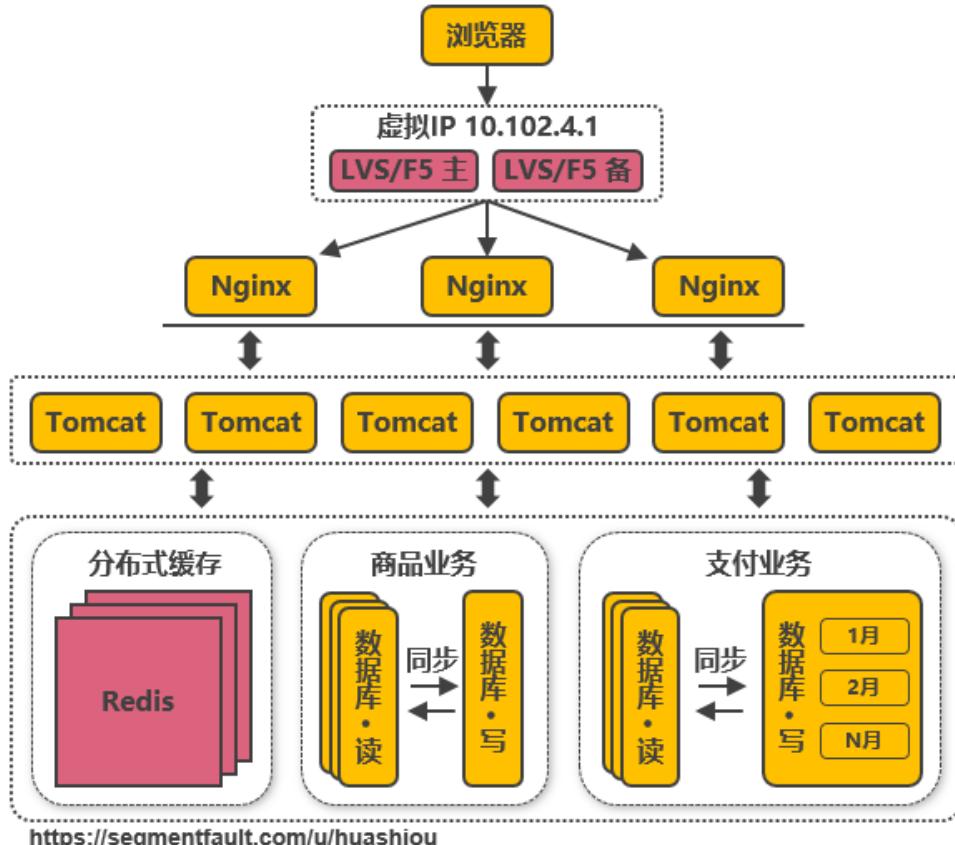
- Using master/slave database to break the database bottleneck

Stage 5 & 6

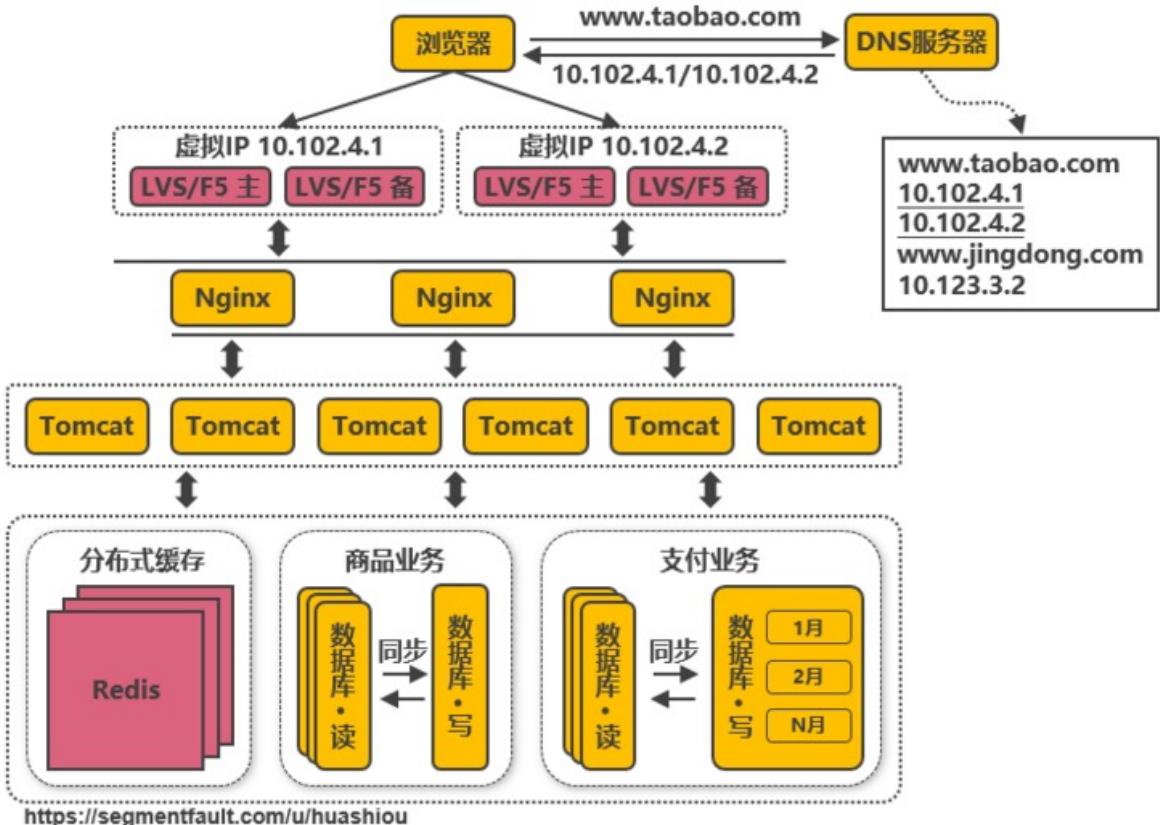


- It is DBA (database admin) time, split databases and tables
- Nginx will meet its limit, around 30000 concurrent users.

Stage 7 & 8



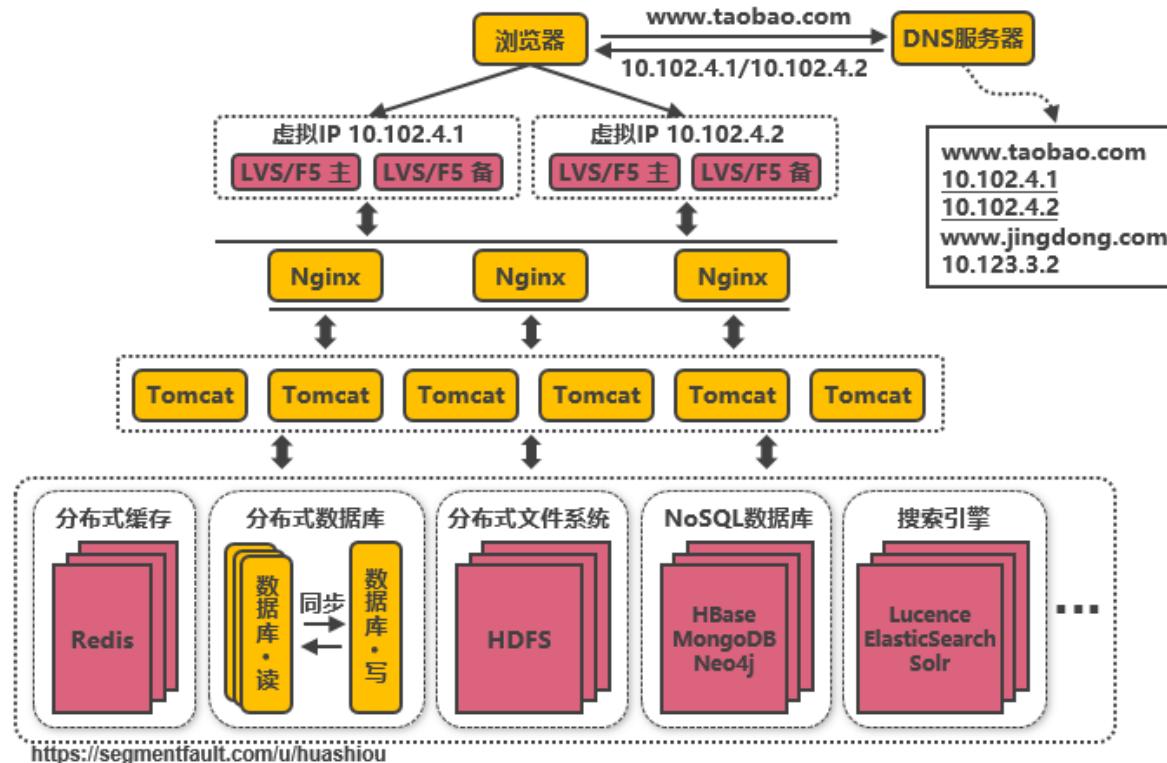
- Level 4 load balance
- Let's say 100k now



- DNS to have many clusters

https://mp.weixin.qq.com/s?_biz=MjM5ODYxMDA5OQ==&mid=2651959595&idx=1&sn=5f0633afd24c547b895f29f6538baa99&scene=1&srcid=0918W9XezgcVJt1eMtOeUE6C#wechat_redirect

Stage 9



key	value
前	静夜思 望庐山瀑布
月	静夜思 月下独酌

“前”字可以索引到两首诗，“月”字也可以索引到两首诗。

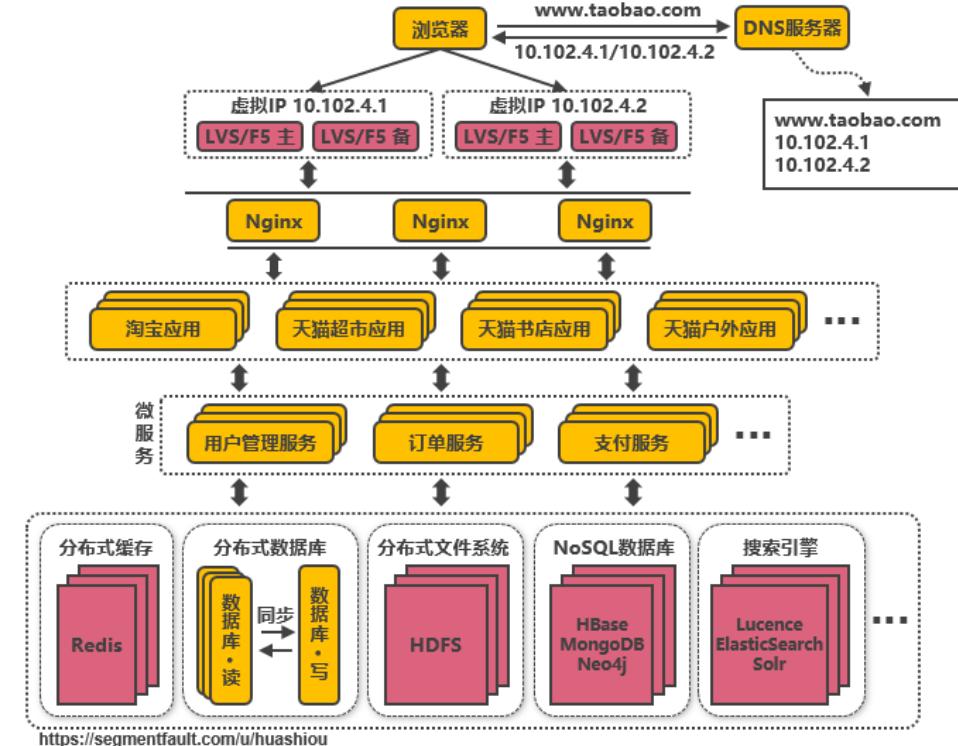
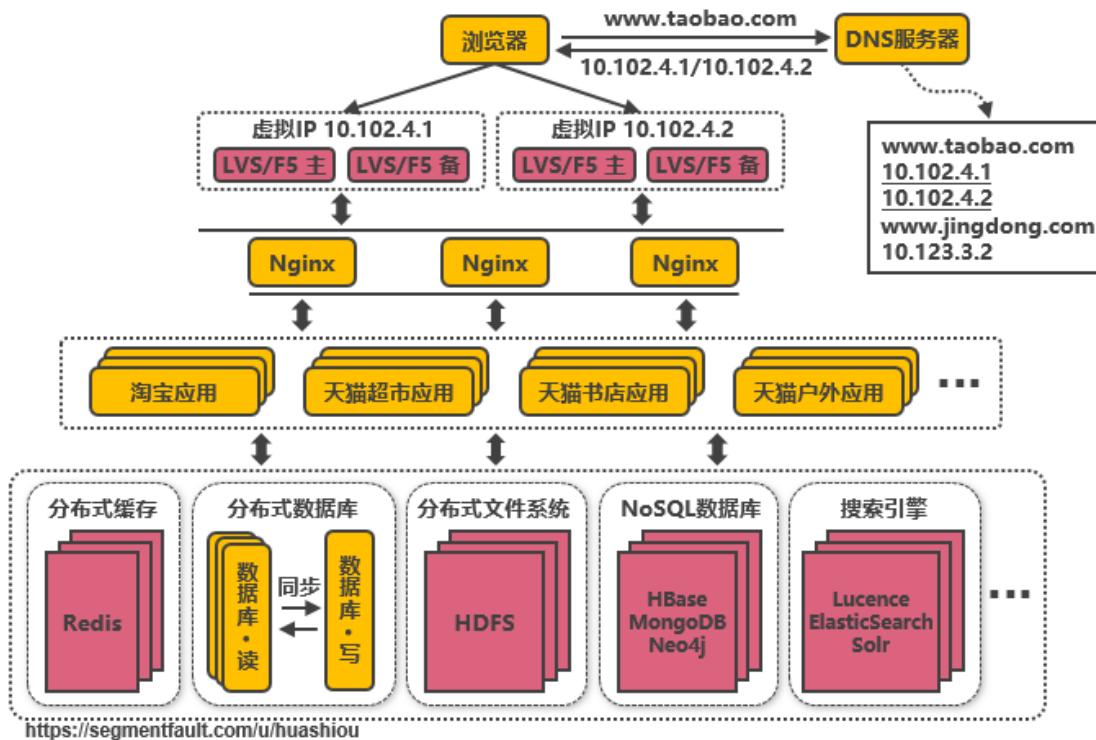


- Database has index function but not professional on that
- Professional search engine (like Elasticsearch) can increase the search efficiency

<https://segmentfault.com/a/1190000018626163>

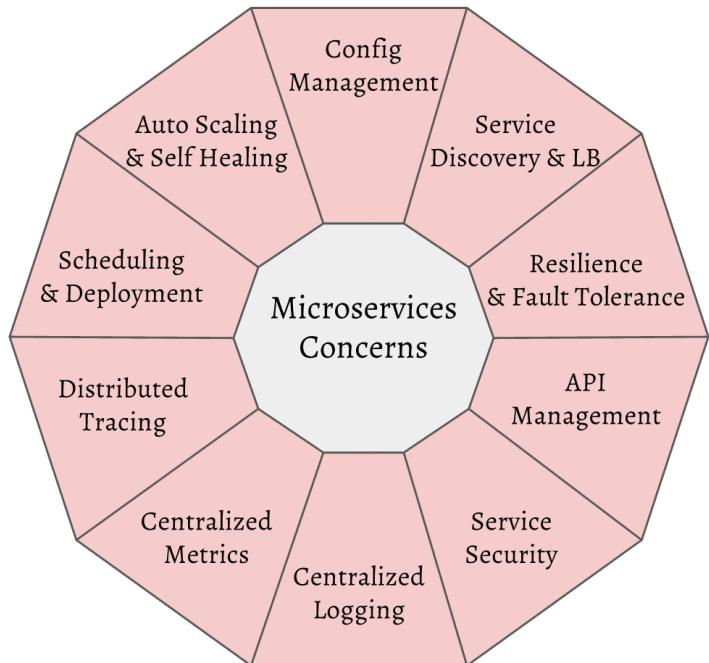
https://blog.csdn.net/m0_46219348/article/details/121153126

Stage 10 & 11



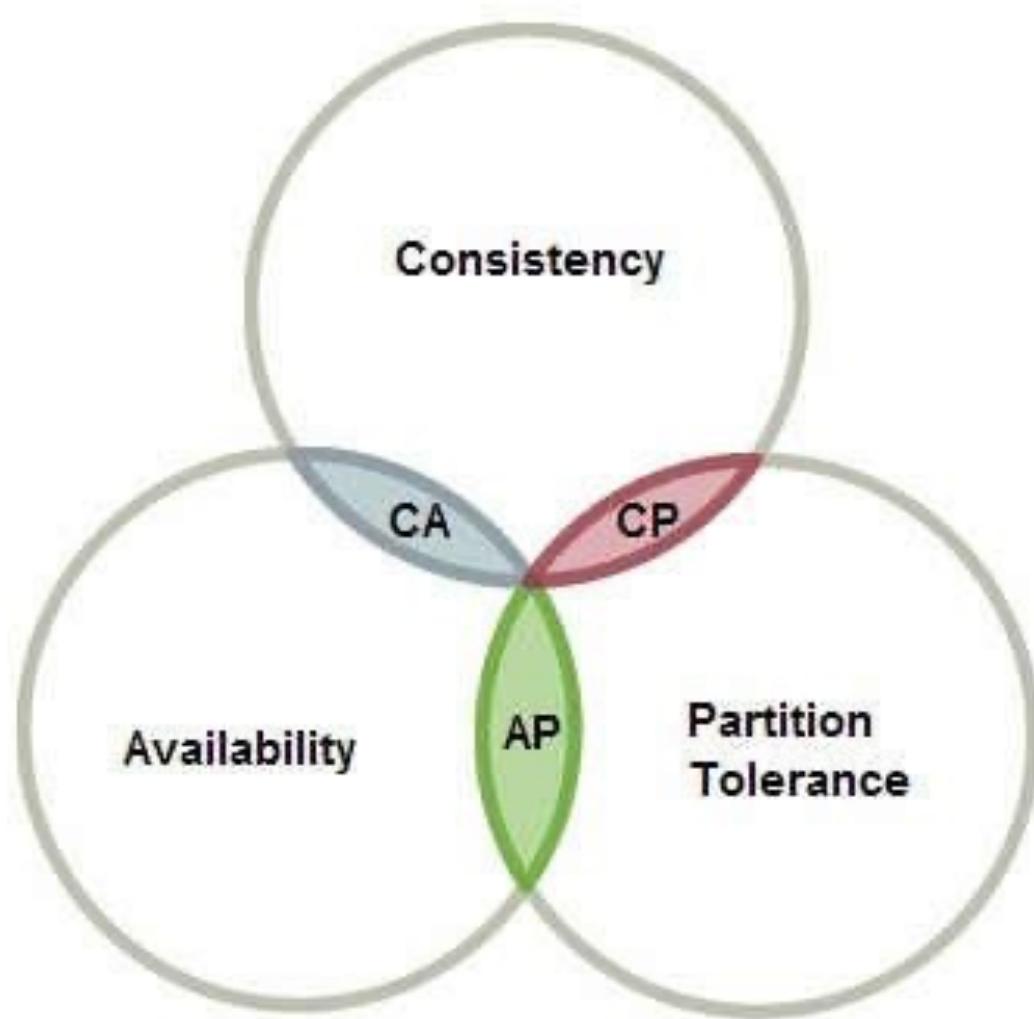
- **Split the application**
- **Middle platform : Build the middle-layer**

MSA - Microservice Architecture



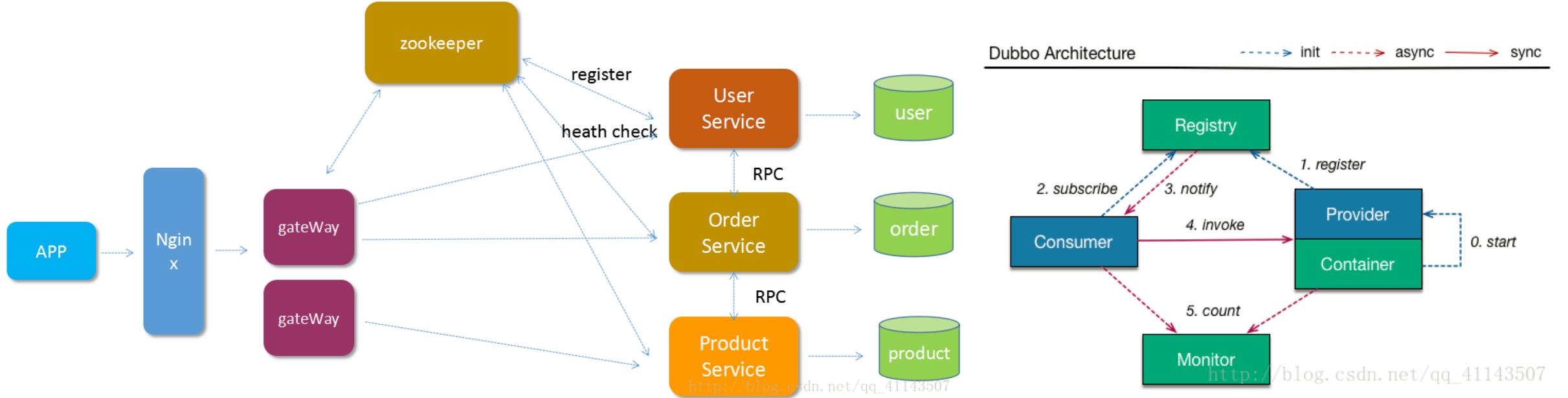
Microservices Concern	Spring Cloud & Netflix OSS	Kubernetes
Configuration Management	Config Server, Consul, Netflix Archaius	Kubernetes ConfigMap & Secrets
Service Discovery	Netflix Eureka, Hashicorp Consul	Kubernetes Service & Ingress Resources
Load Balancing	Netflix Ribbon	Kubernetes Service
API Gateway	Netflix Zuul	Kubernetes Service & Ingress Resources
Service Security	Spring Cloud Security	-
Centralized Logging	ELK Stack (Logstash)	EFK Stack (Fluentd)
Centralized Metrics	Netflix Spectator & Atlas	Heapster, Prometheus, Grafana
Distributed Tracing	Spring Cloud Sleuth, Zipkin	OpenTracing, Zipkin
Resilience & Fault Tolerance	Netflix Hystrix, Turbine & Ribbon	Kubernetes Health Check & resource isolation
Auto Scaling & Self Healing	-	Kubernetes Health Check, Self Healing, Autoscaling
Packaging, Deployment & Scheduling	Spring Boot	Docker/Rkt, Kubernetes Scheduler & Deployment
Job Management	Spring Batch	Kubernetes Jobs & Scheduled Jobs
Singleton Application	Spring Cloud Cluster	Kubernetes Pods

CAP – Only can choose two



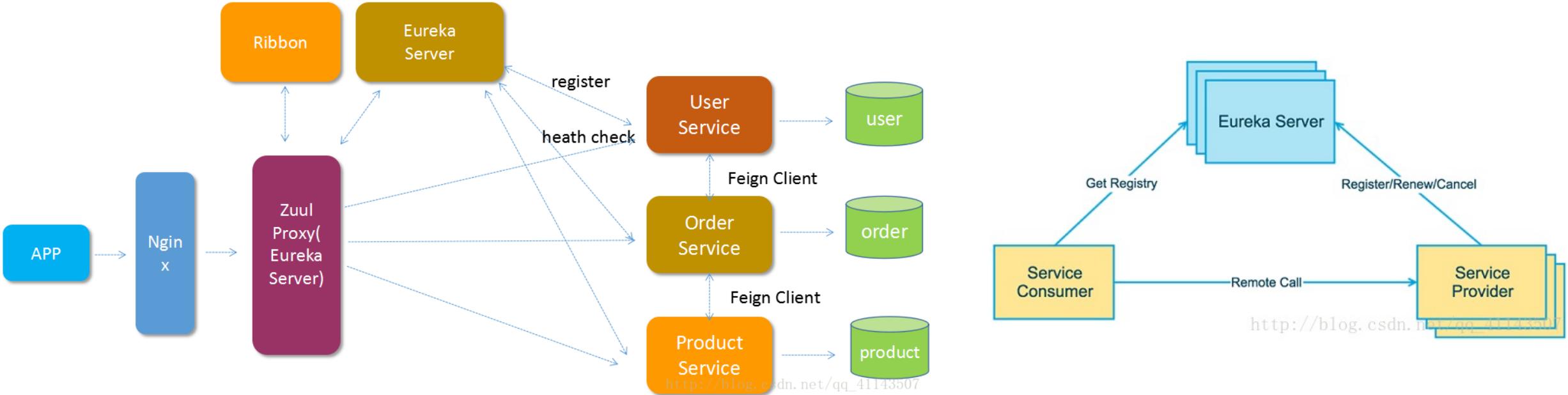
<https://zhuanlan.zhihu.com/p/50990721>

Microservice - Dubbo



- **Created by Alibaba, contributed to Apache**
- **Based on RPC, still run in Tomcat**

Microservice - Spring cloud



- It is from Netflix and be famous “Spring” with mature community
- Based on https
- Spring boot is a new way to pack each service with a web server

https://www.sohu.com/a/206889245_115128

<https://dzone.com/articles/microservice-architecture-with-spring-cloud-and-do>

Spring cloud – Service discovery

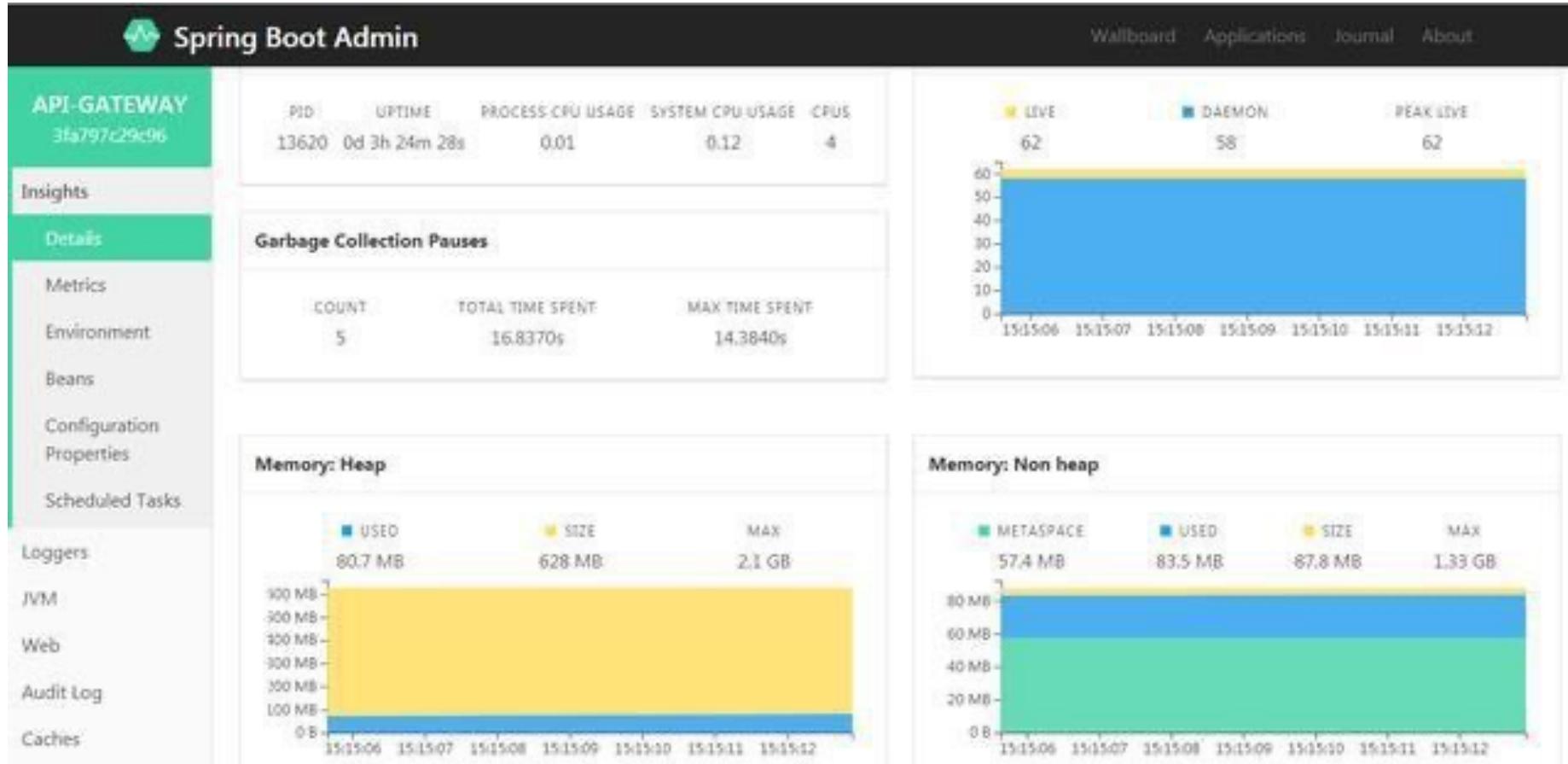
```
0  import org.springframework.boot.autoconfigure.condition.ConditionalOnProperty;
1  import org.springframework.cloud.config.client.ConfigServicePropertySourceLocator;
2  import org.springframework.cloud.netflix.eureka.EurekaClientAutoConfiguration;
3  import org.springframework.cloud.netflix.eureka.EurekaDiscoveryClientConfiguration;
4  import org.springframework.context.annotation.Configuration;
5  import org.springframework.context.annotation.Import;
6
7  /**
8   * Eureka-specific helper for config client that wants to lookup the config server via
9   * discovery.
10  *
11  * @author Dave Syer
12  */
13 @ConditionalOnClass(ConfigServicePropertySourceLocator.class)
14 @ConditionalOnProperty(value = "spring.cloud.config.discovery.enabled", matchIfMissing = false)
15 @Configuration
16 @Import({ EurekaDiscoveryClientConfiguration.class, // this emulates @EnableDiscoveryClient, the import selector
17           EurekaClientAutoConfiguration.class })
18 public class EurekaDiscoveryClientConfigServiceBootstrapConfiguration {
19 }
```

通过SpringCloud Config去管理eureka的配置文件

<https://blog.csdn.net/yang131peng>

Need to write the discovery in the code

Spring cloud



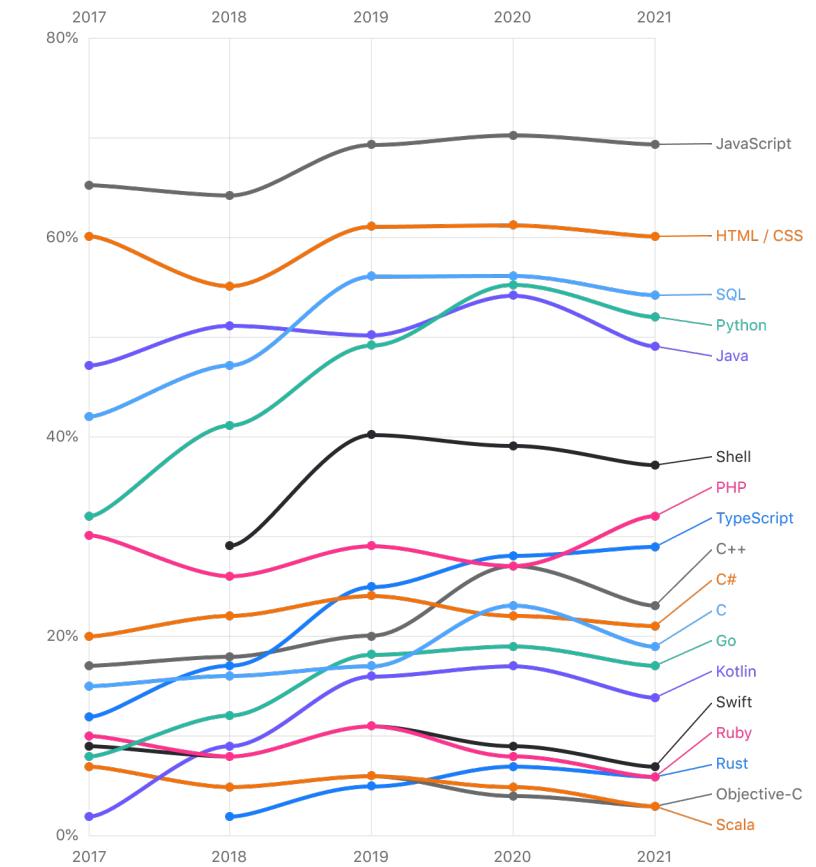
<https://programmer.help/blogs/springcloud-service-registration-and-discovery.html>
<https://baijiahao.baidu.com/s?id=1640123723252607826&wfr=spider&for=pc>

Dubbo vs. Spring cloud

核心要素	Dubbo	Spring Cloud
服务注册中心	Zookeeper、Redis	Spring Cloud Netflix Eureka
服务调用方式	RPC	REST API
服务网关	无	Spring Cloud Netflix Zuul
断路器	不完善	Spring Cloud Netflix Hystrix
分布式配置	无	Spring Cloud Config
分布式追踪系统	无	Spring Cloud Sleuth
消息总线	无	Spring Cloud Bus
数据流	无	Spring Cloud Stream 基于Redis,Rabbit,Kafka实现的消息微服务
批量任务	无	Spring Cloud Task http://blog.csdn.net/qq_41143507

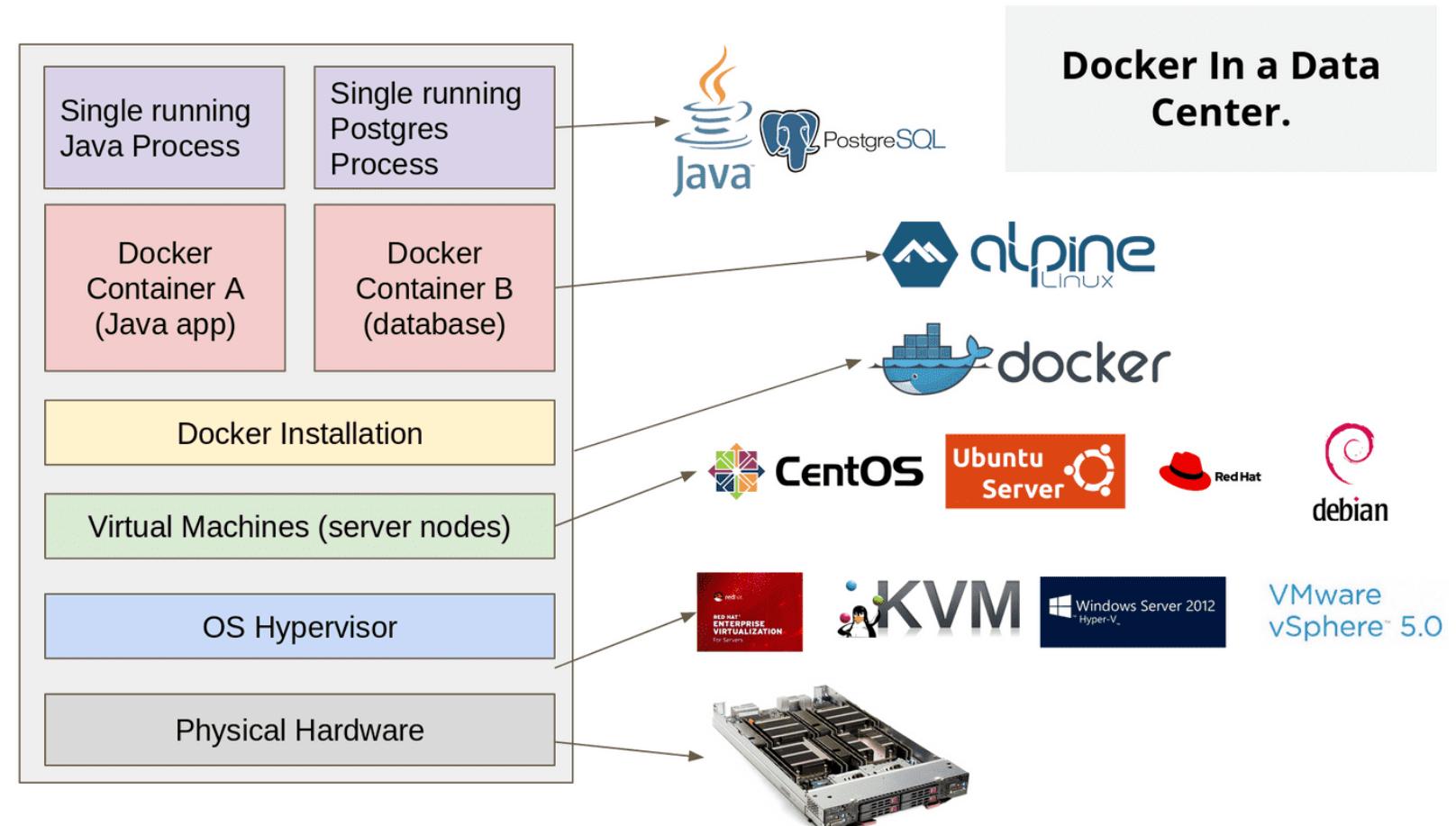
线程数	Dubbo	Spring Cloud
10线程	2.75	6.52
20线程	4.18	10.03
50线程	10.3	28.14
100线程	20.13	55.23
200线程	42	110.21

Java, favored by big project



<https://blog.csdn.net/DeveloperFront/article/details/108924354>
<https://www.jetbrains.com/lp/devcosystem-2021/>

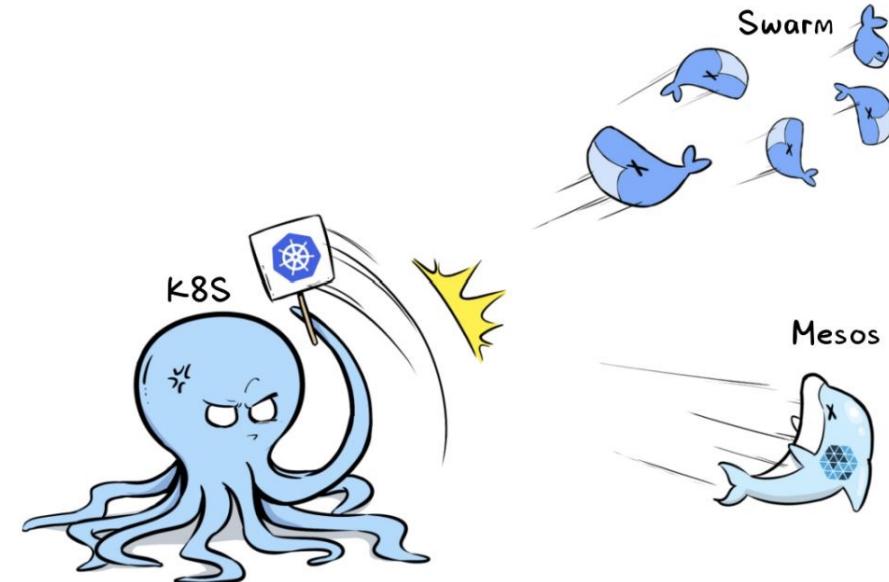
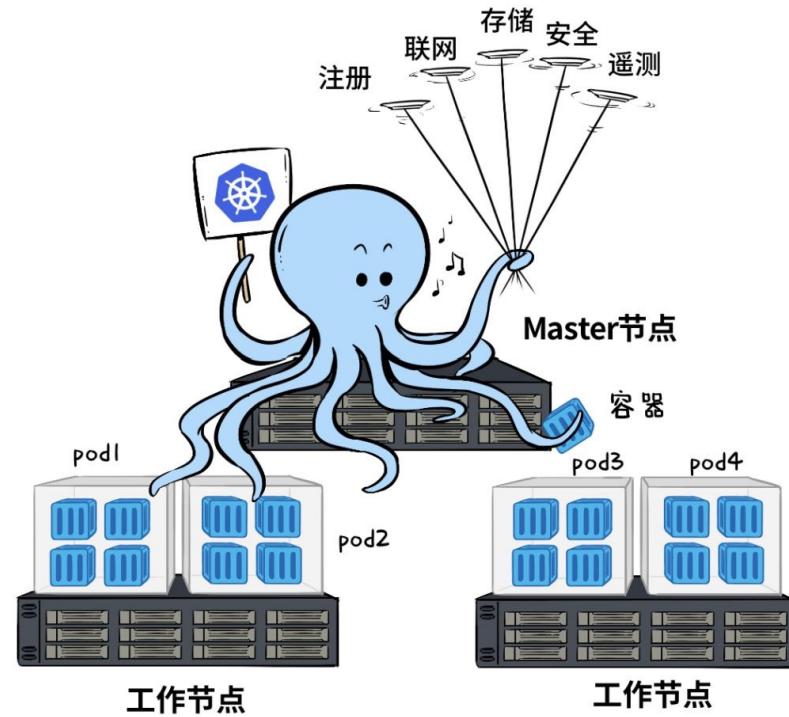
Why Docker?



- ❑ Docker: pack the application with environment (dependency) together.

<https://www.iteachrecruiters.com/blog/docker-explained-visually-for-non-technical-folks/>

Docker orchestration – k8s



Kubernetes: also supports **service discovery and load balancing**
supports **Horizontal scaling**

<https://kubernetes.io/>

<https://www.zhihu.com/question/329365548>

Spring cloud vs. k8s



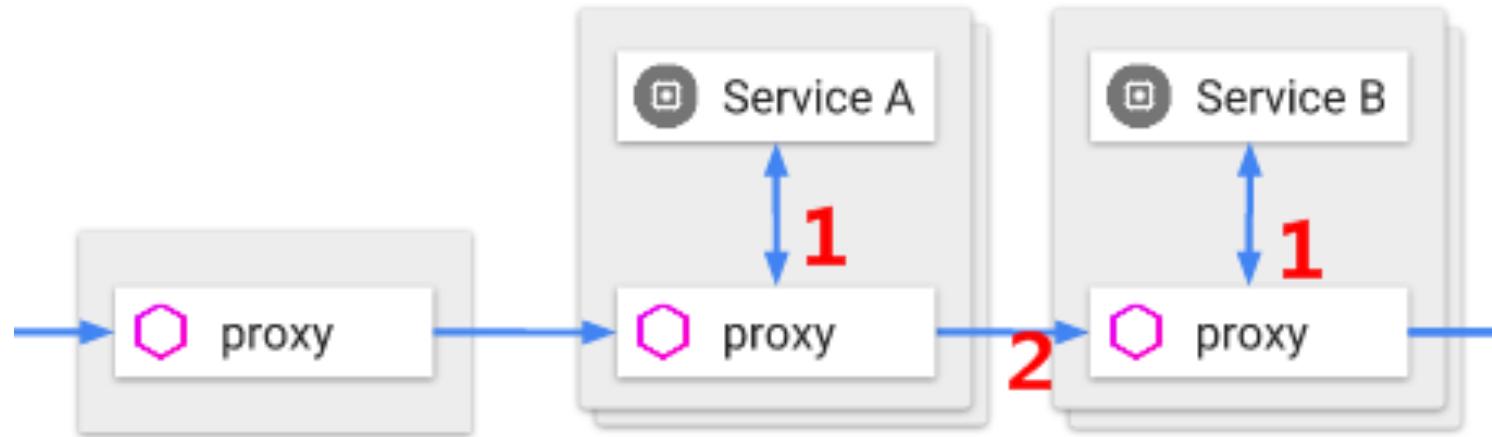
<https://cloud.tencent.com/developer/article/1776313>

Spring cloud vs. k8s

The diagram illustrates the relationship between three layers of cloud infrastructure: Spring Cloud, Kubernetes, and IaaS. A vertical stack of boxes on the left lists various cloud-native features. Brackets on the right group these features into three categories: **Kubernetes** (top), **Spring Cloud** (middle), and **IaaS** (bottom). A central table maps each feature to its corresponding implementation in **Spring Cloud with Kubernetes**.

Capability	Spring Cloud with Kubernetes
DevOps Experience	Self service, multi-environment capabilities
Auto Scaling & Self Healing	Pod/Cluster Autoscaler, HealthIndicator, Scheduler
Resilience & Fault Tolerance	HealthIndicator, Hystrix, HealthCheck, Process Check
Distributed Tracing	Zipkin
Centralized Metrics	Heapster, Prometheus, Grafana
Centralized Logging	EFK
API Gateway	Spring Batch, Scheduled Job
Job Management	Ribbon, Service
Singleton Application	Service
Load Balancing	Externalized Configurations, ConfigMap, Secret
Service Discovery	Apache Camel, Spring Framework
Configuration Management	Spring Boot maven plugin
Application Packaging	Deployment strategy, A/B, Canary, Scheduler strategy
Deployment & Scheduling	Docker, Pods
Process Isolation	Namespaces, Authorizations
Environment Management	CPU and memory limits, Namespace resource quotas
Resource Management	GCE, Azure, CenturyLink, VMware, Openstack
Operating System	
Virtualization	
Hardware, Storage, Networking	

Istio - non-intrusive



Istio brings standard, universal traffic management, telemetry, and security to complex deployments

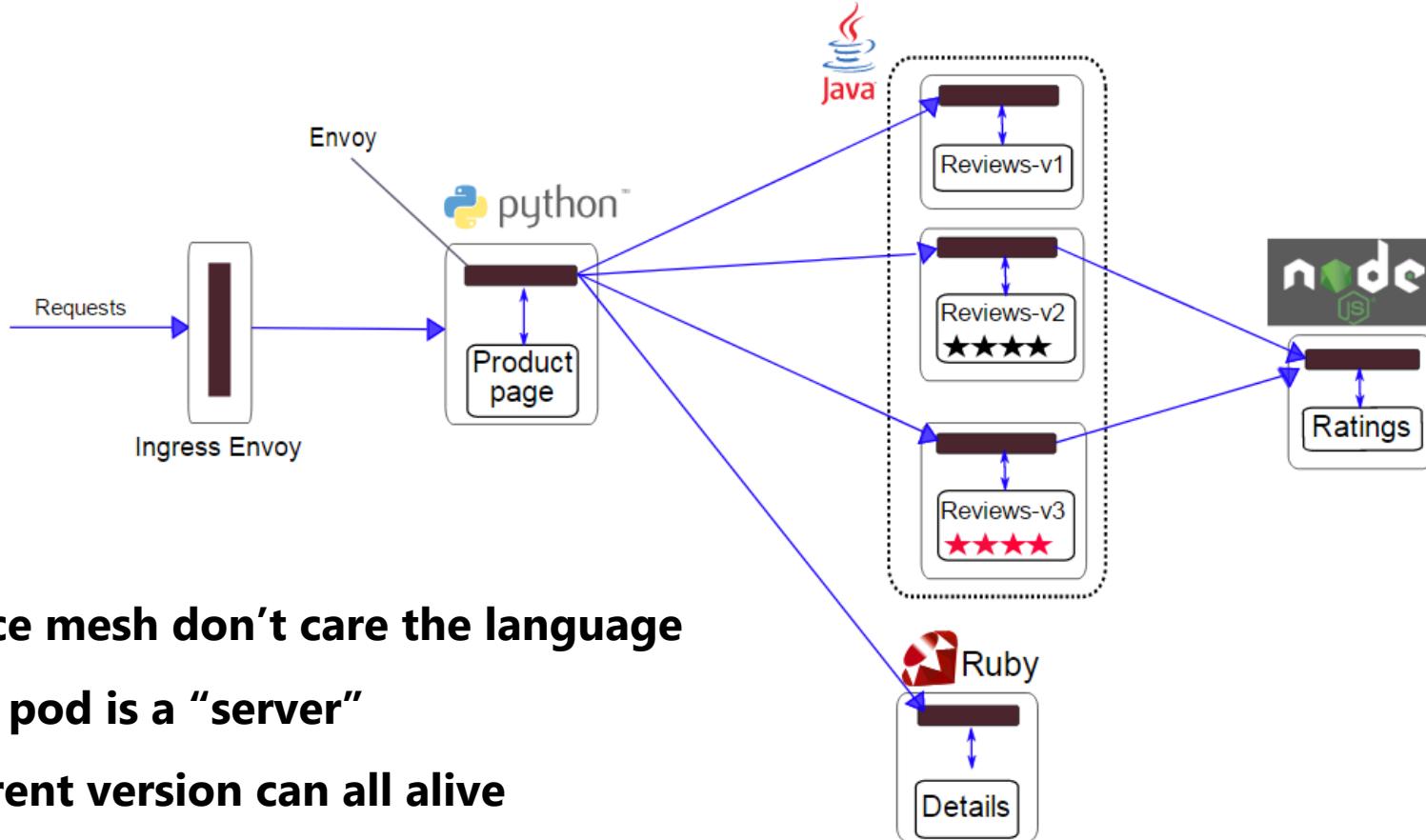
<https://blog.51cto.com/zero01/2569374>

<http://xiaorui.cc/archives/6051>

https://www.sohu.com/a/319050261_178889

<https://zhuanlan.zhihu.com/p/80265643>

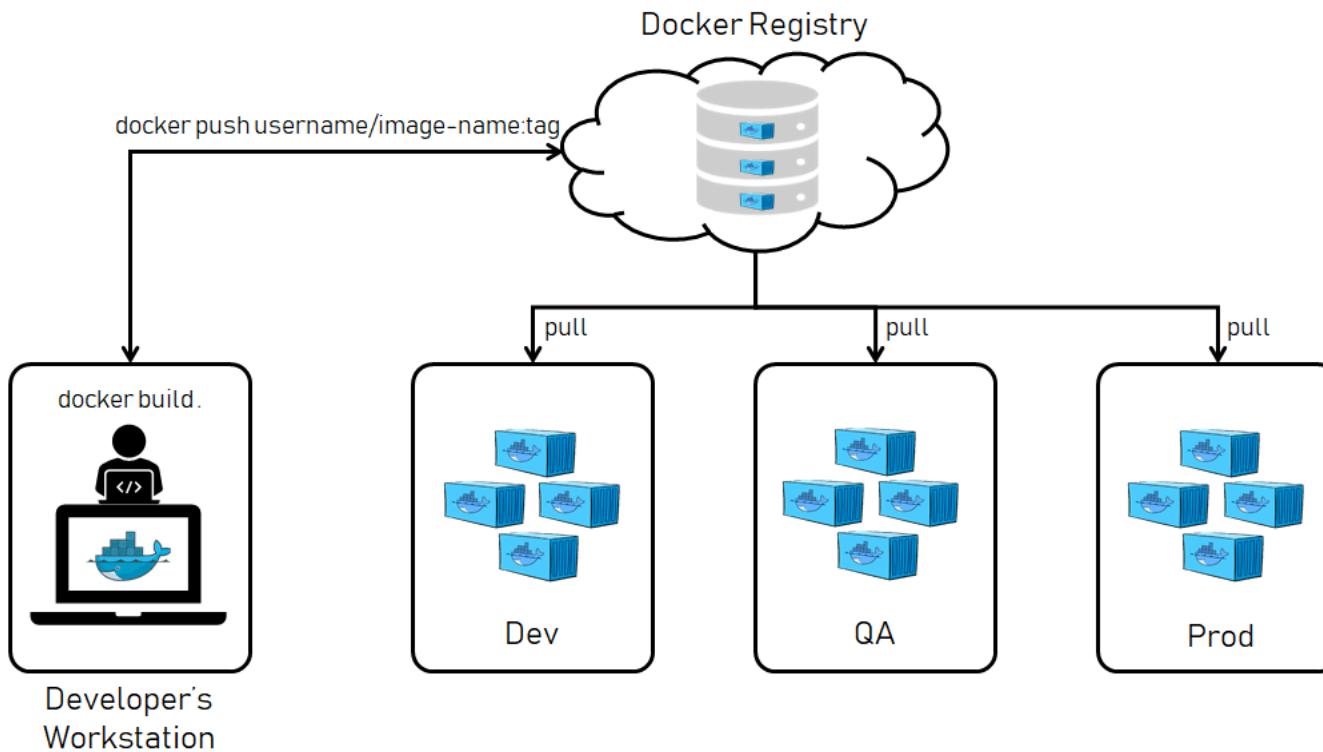
Service mesh by istio



https://blog.csdn.net/qq_37950254/article/details/89603147

<https://developer.ibm.com/patterns/manage-microservices-traffic-using-istio/>

Jenkins - Docker Build and Publish

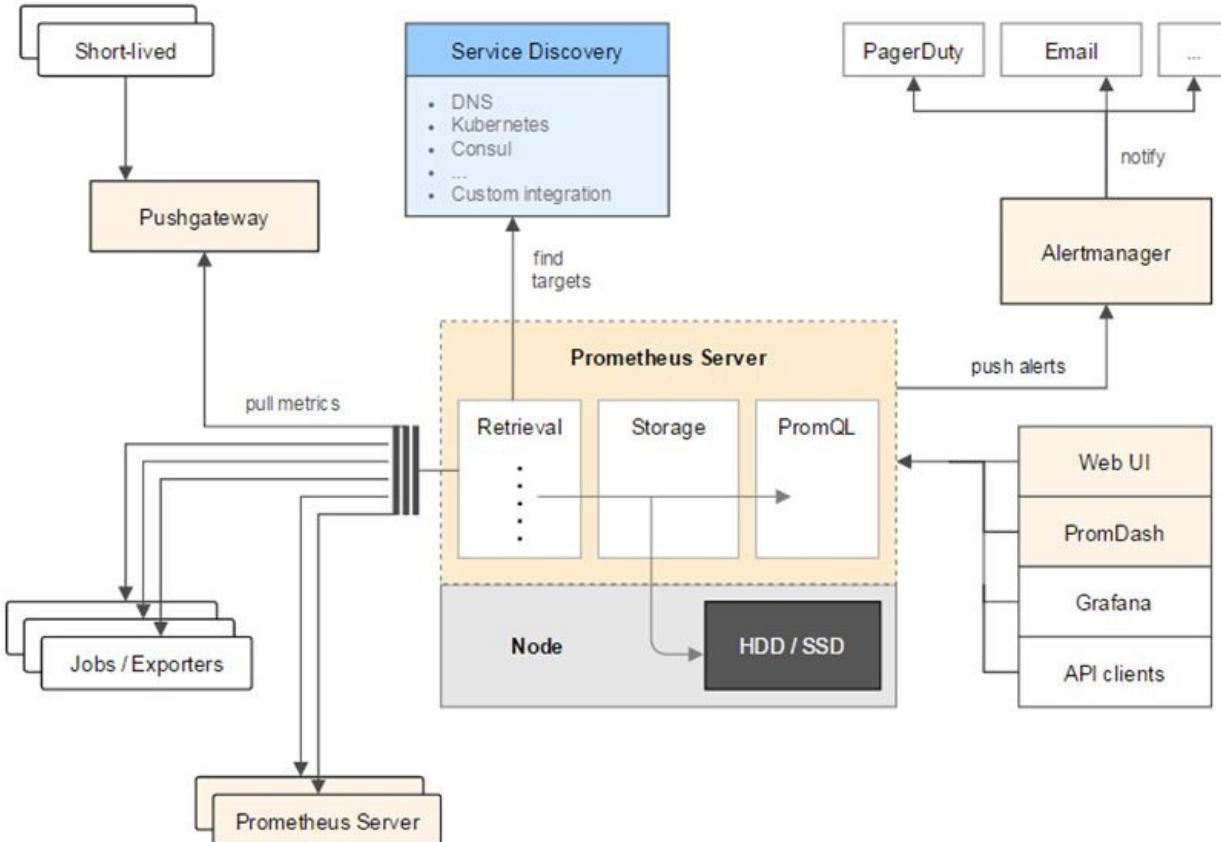


- Auto build/publish tool
- Can be applied to many programs including docker.
- Support sidecars

<https://www.jenkins.io/>

<https://digitalvarys.com/docker-build-and-publish-with-jenkins-dsl-script/>

Prometheus – monitor the mesh



- Collect performance data
- Trigger alert on pre-defined condition
- On alert, auto-scale the dockers

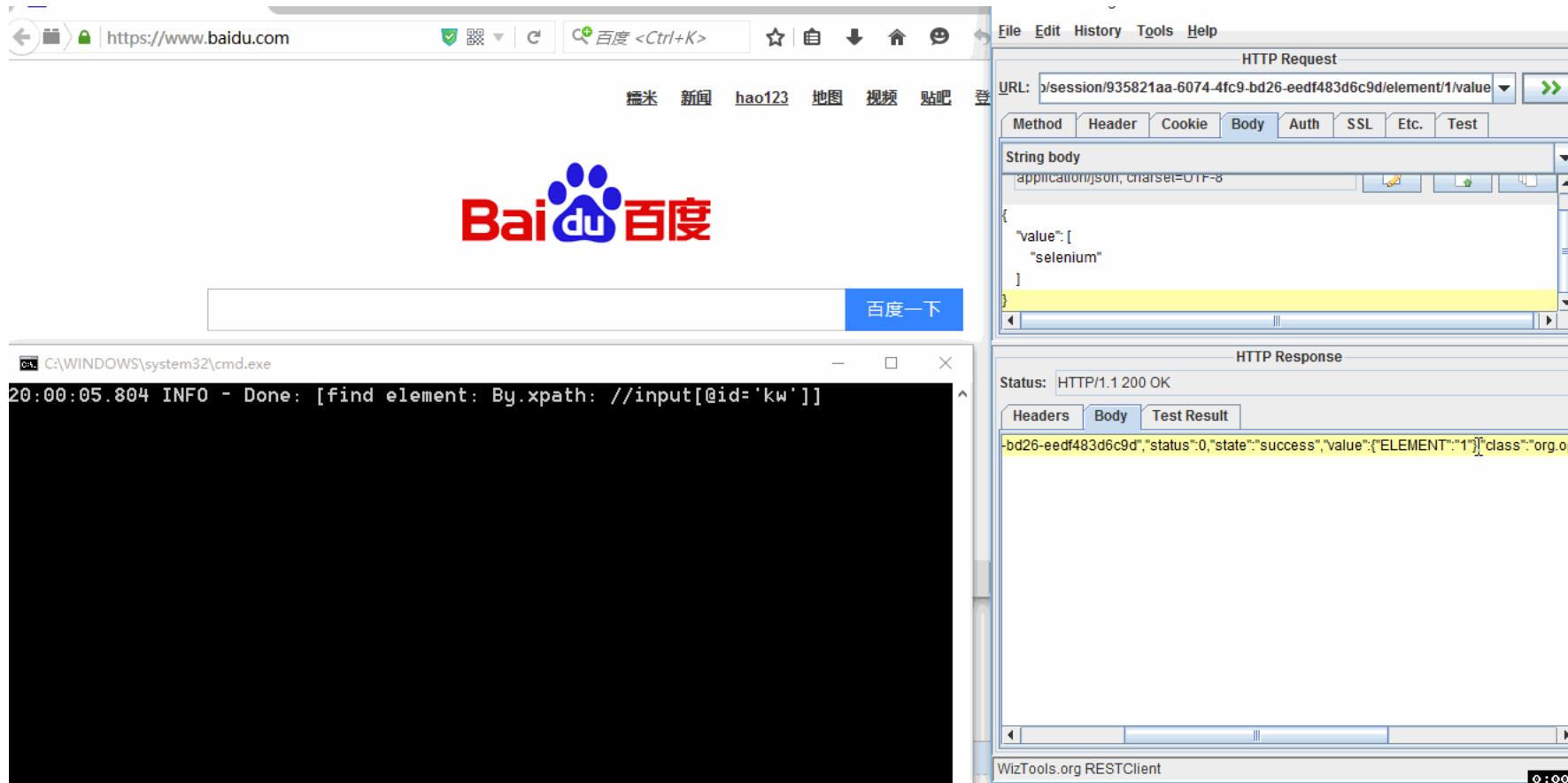
Don't forget **TEST**



- Test plan is **VERY** important
- Test case need to cover all possible conditions, especially kind of "stupid" things.

<https://www.esds.co.in/blog/manual-testing-process-lifecycle/>

UI test tool - selenium



- Selenium can operate the browser by code.

<http://www.51testing.com/html/91/n-3959391.html>

API test tool - postman



The screenshot shows the Postman application interface. On the left, the sidebar displays collections like 'Postman Echo' and 'Report', and requests for 'GET' and 'POST'. A specific 'GET' request for 'getSingleChatInfoNew' is selected in the center. The request details show a GET method and URL: `http://172.28.16.23:8081/inner/getSingleChatInfoNew?authSign=C8363025-BDFE-409F-8C08-8DAE`. The 'Tests' tab is highlighted with a red box and arrow, containing the following script:

```
1 if (responseCode.code === 200) {  
2     var body = JSON.parse(responseBody);  
3     tests["Body contains code 0"] = 0 === body.meta.code;  
4 }  
5  
6 }
```

Below the tests, the response body is shown in JSON format:

```
1 {  
2     "meta": {  
3         "code": 0,  
4         "message": "success"  
5     },  
6     "data": {  
7         "ChatContents": [  
8             {  
9                 "feedInfo": {  
10                     "feedId": "c_9150630535344410",  
11                     "cardNo": "254771",  
12                     "content": "I am I"  
13                 }  
14             }  
15         ]  
16     }  
17 }
```

- Simulate the get/post data can show the response

<https://www.esds.co.in/blog/manual-testing-process-lifecycle/>

So many technologies

The screenshot shows a web browser window with the URL [alibabacloud.com](https://www.alibabacloud.com) in the address bar. The page content is organized into several sections:

- Elastic Computing**
- Networking & CDN**
- Database**
- Storage**
- Security**
- Enterprise Applications & Cloud Communication**
- Analytics**
- Artificial Intelligence**
- Media Services**
- Hybrid Cloud**
- Container & Middleware**
- Developer Services**
- Internet of Things**
- Alibaba Cloud Academy**

Database →

Relational Database

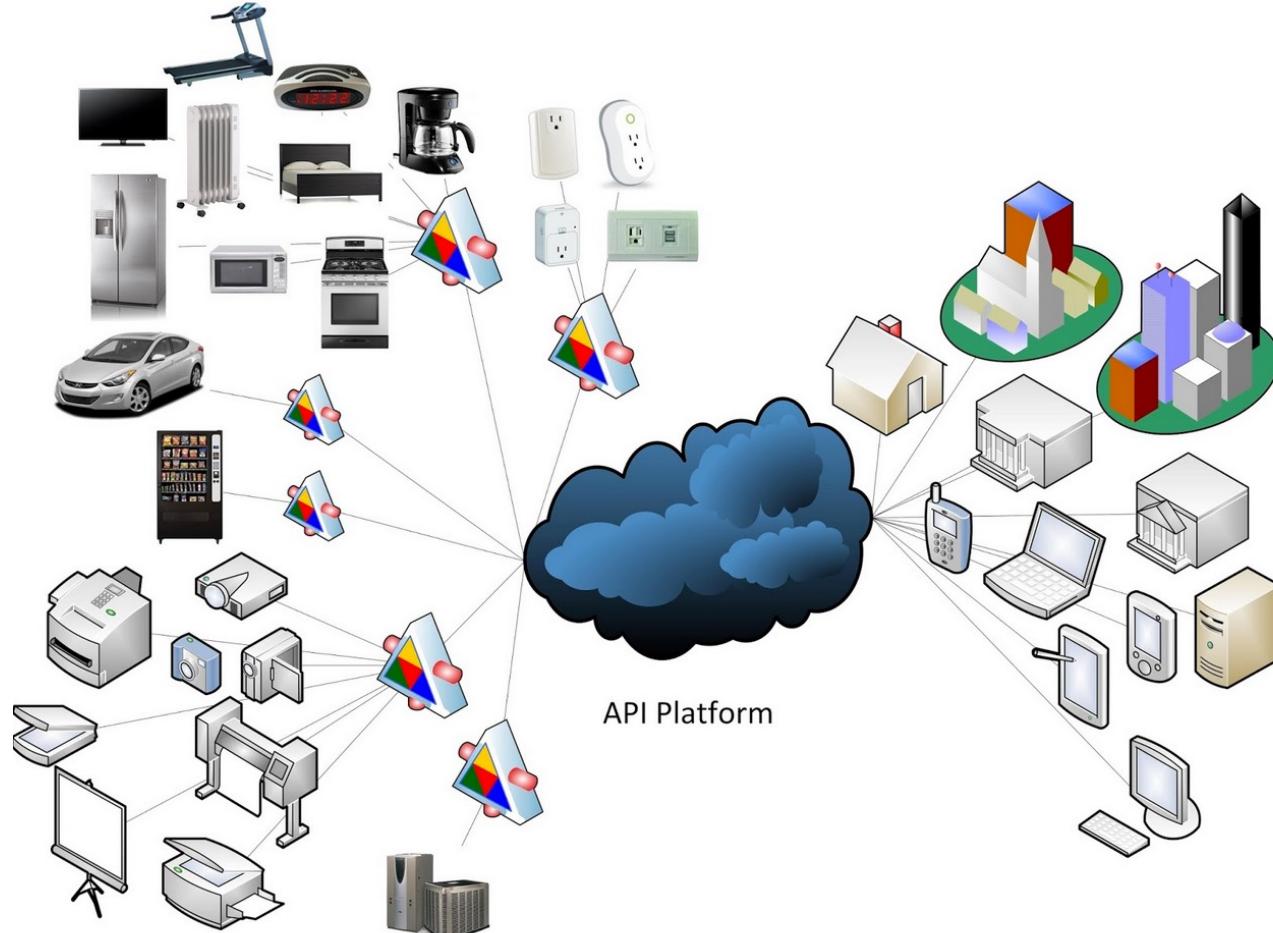
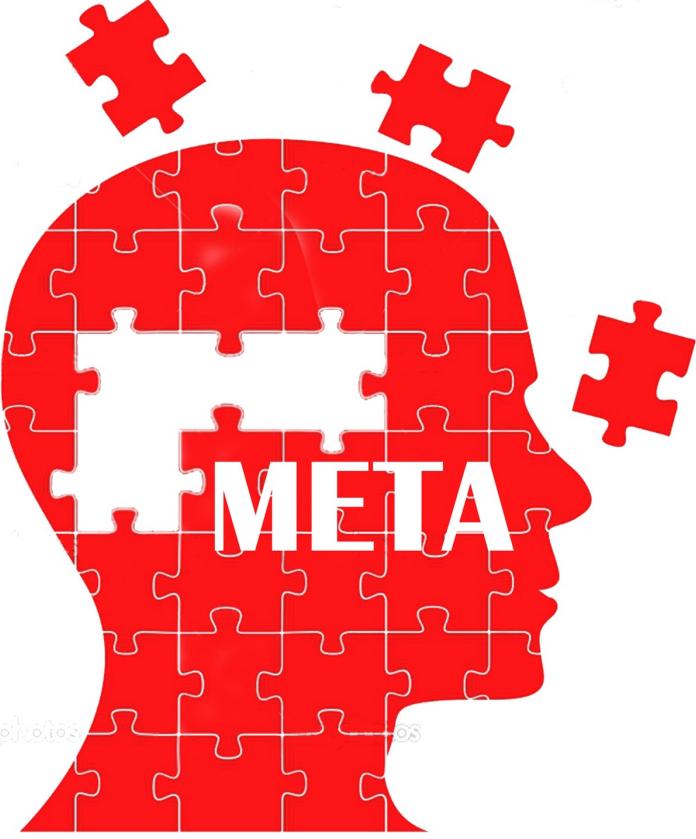
PolarDB A next-generation relational database	PolarDB-X Be scalable for big relational databases
ApsaraDB RDS for MySQL Featured stable and capable with MySQL	ApsaraDB RDS for PostgreSQL Get low latency with high concurrency
ApsaraDB RDS for SQL Server Flexible version control and lower costs	ApsaraDB RDS for MariaDB TX Capable and fully managed database
DBStack ApsaraDB platform deployed on any infrastructure, all under your ownership	ApsaraDB for OceanBase Financial-grade database: high stability, high scalability, and high performance

Dedicated Cluster

BIGGEST SALE \$3.5/year

<https://www.alibabacloud.com/product>

Everything ONLINE age



More **smart things** than human beings online!