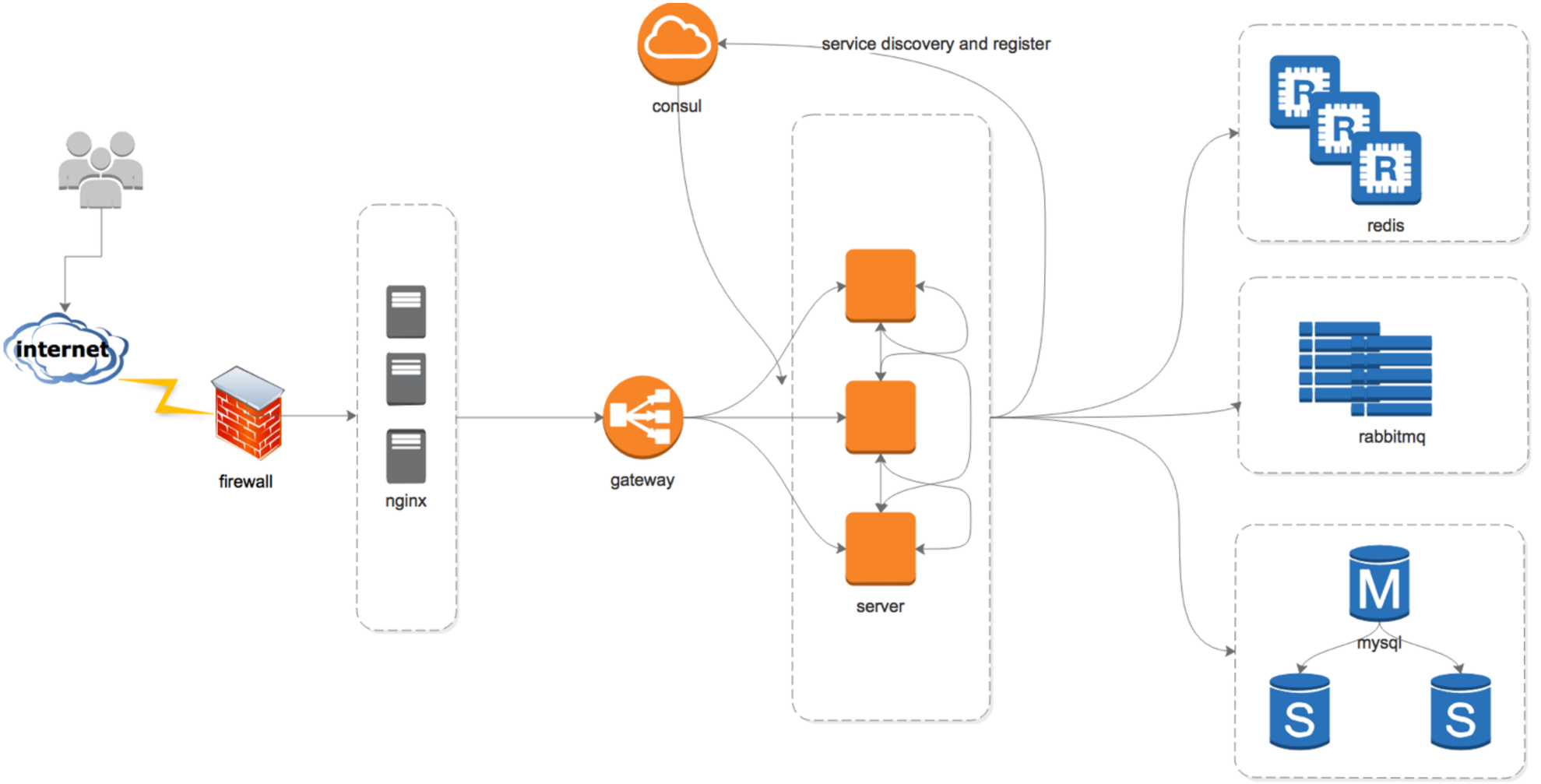
如何构建高可用的分布式系统

LIAOKAILIN

分布式要素


1
响应时间，并发数，吞吐量
性能
高可用
2
减少系统不可用的时间
3
可以通过添加/减少服务器提高/降低
服务的处理能力
伸缩
可拓展
4
在对现有系统影响较小的情况下，系
统功能可持续拓展、提升的能力
5
https,聚安全,加解密，openid
安全

流程


service discovery and register
consul
internet
firewall
nginx
gateway
server
redis
rabbitmq
mysql
M
S
S

大纲

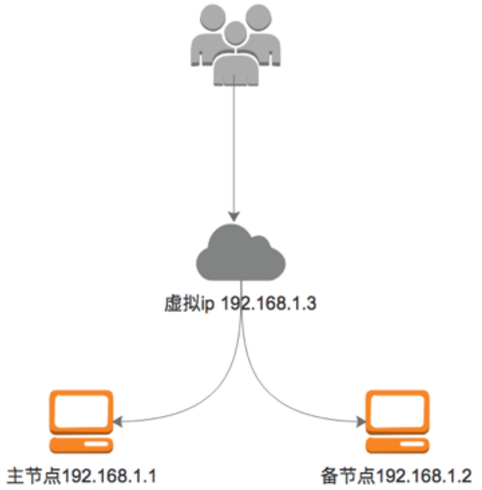
1. 客户端到反向代理
2. 反向代理到Gateway
3. 服务间调用Services
4. 分布式缓存Cache
5. 消息队列MQ
6. 数据库Datebase
7. 监控Monitor
8. 发布与升级

END

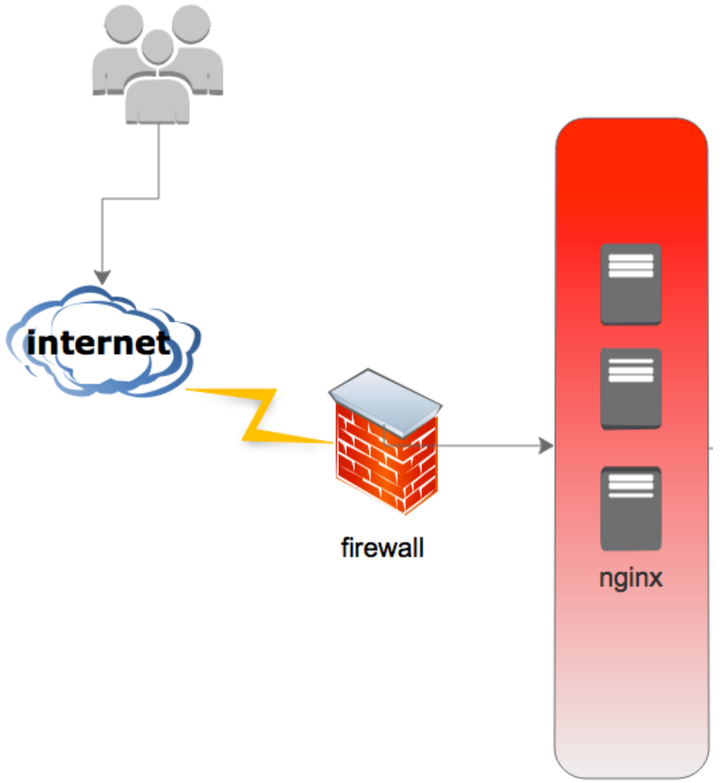
客户端到反向代理

F5实现硬负载

Nginx实现软负载



keepalived+vip



客户端到反向代理

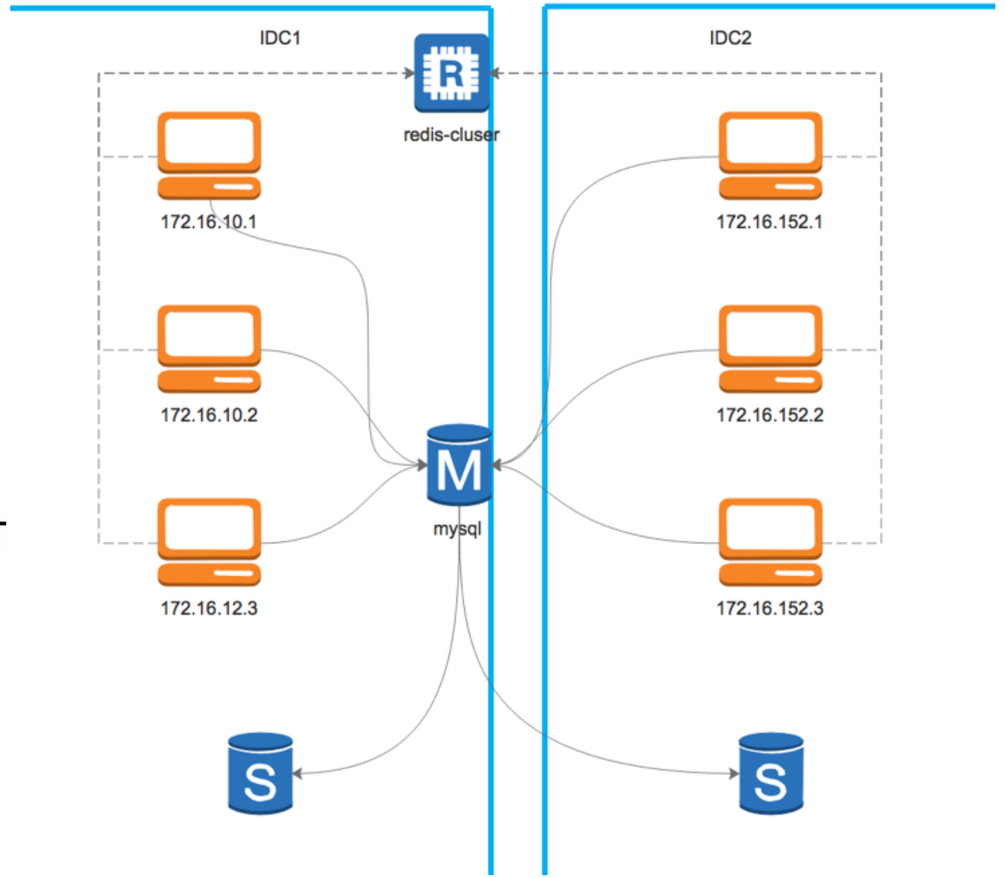
通过nginx配置文件,nginx.conf

```
upstream test {
    server 192.168.10.1:8000;
    server 192.168.10.1:8001;
}
```

同城双活

```
upstream test {
server 10.0.40.126:80 max_fails=1
fail_timeout=30s;
server 10.0.40.124:80 backup;
}
```

当40.126 80服务30秒内出现一次请求失败的时候，启用40.124 backup



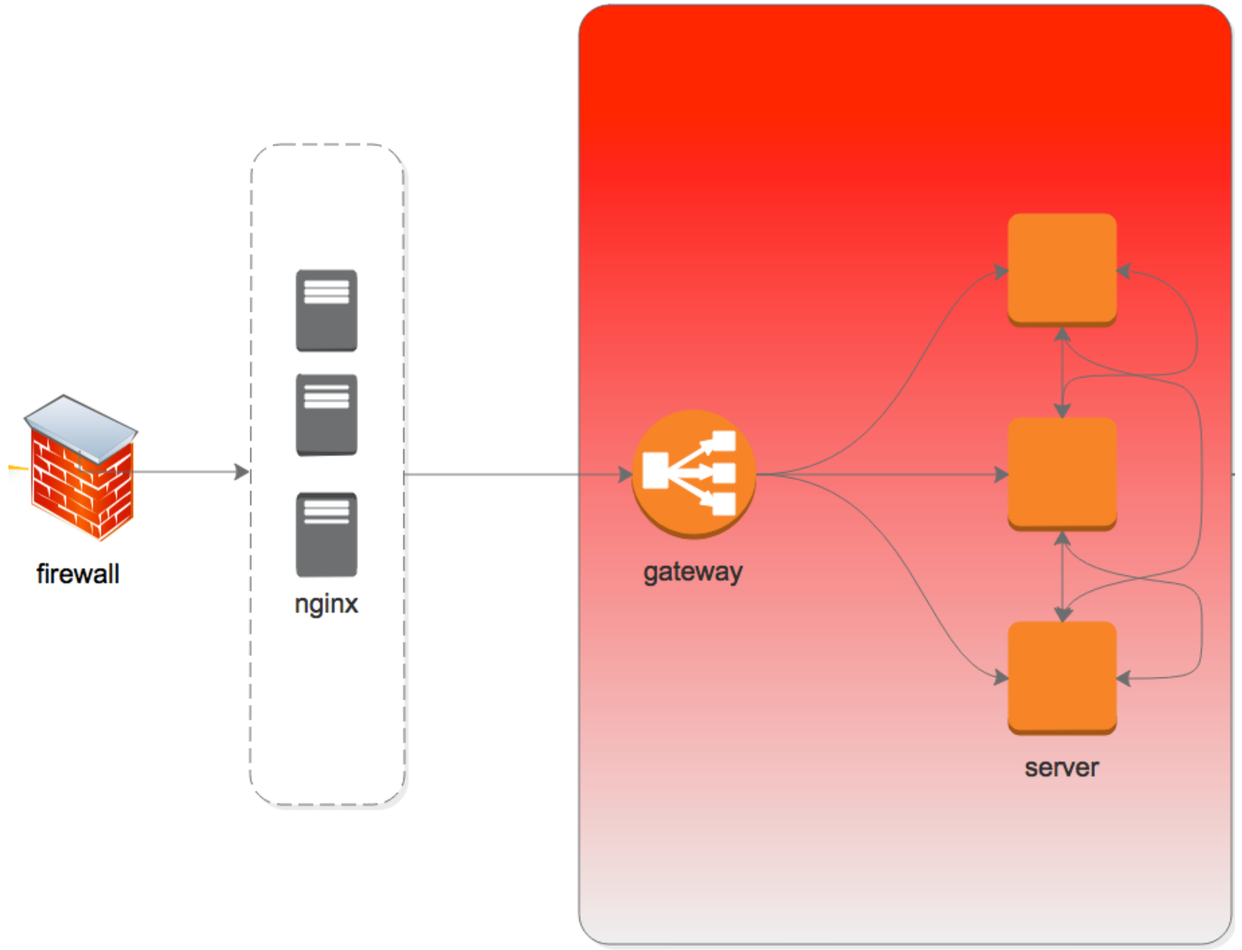
GATEWAY & SERVICES

负载均衡

服务发现

服务选择

服务检测

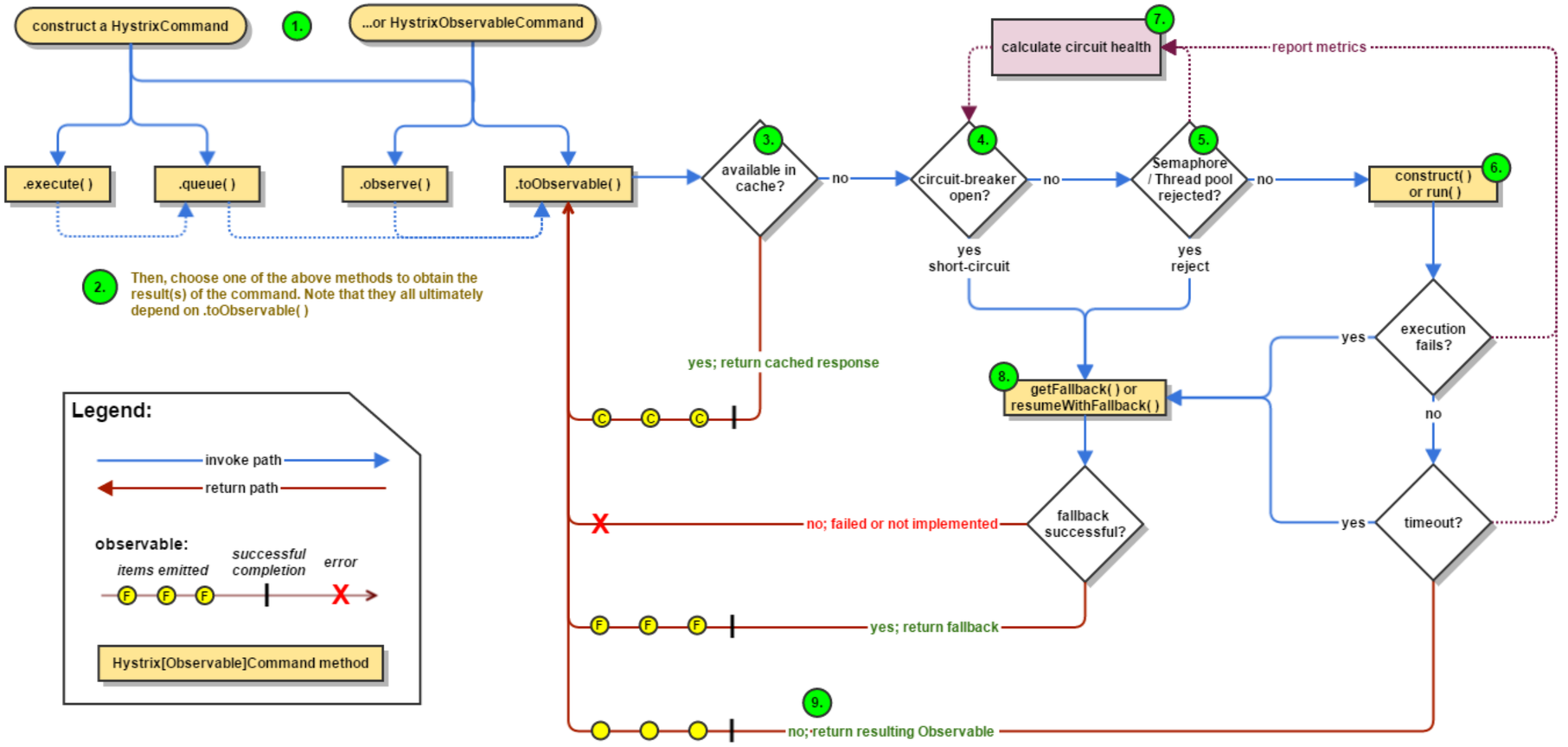


服务保护

熔断

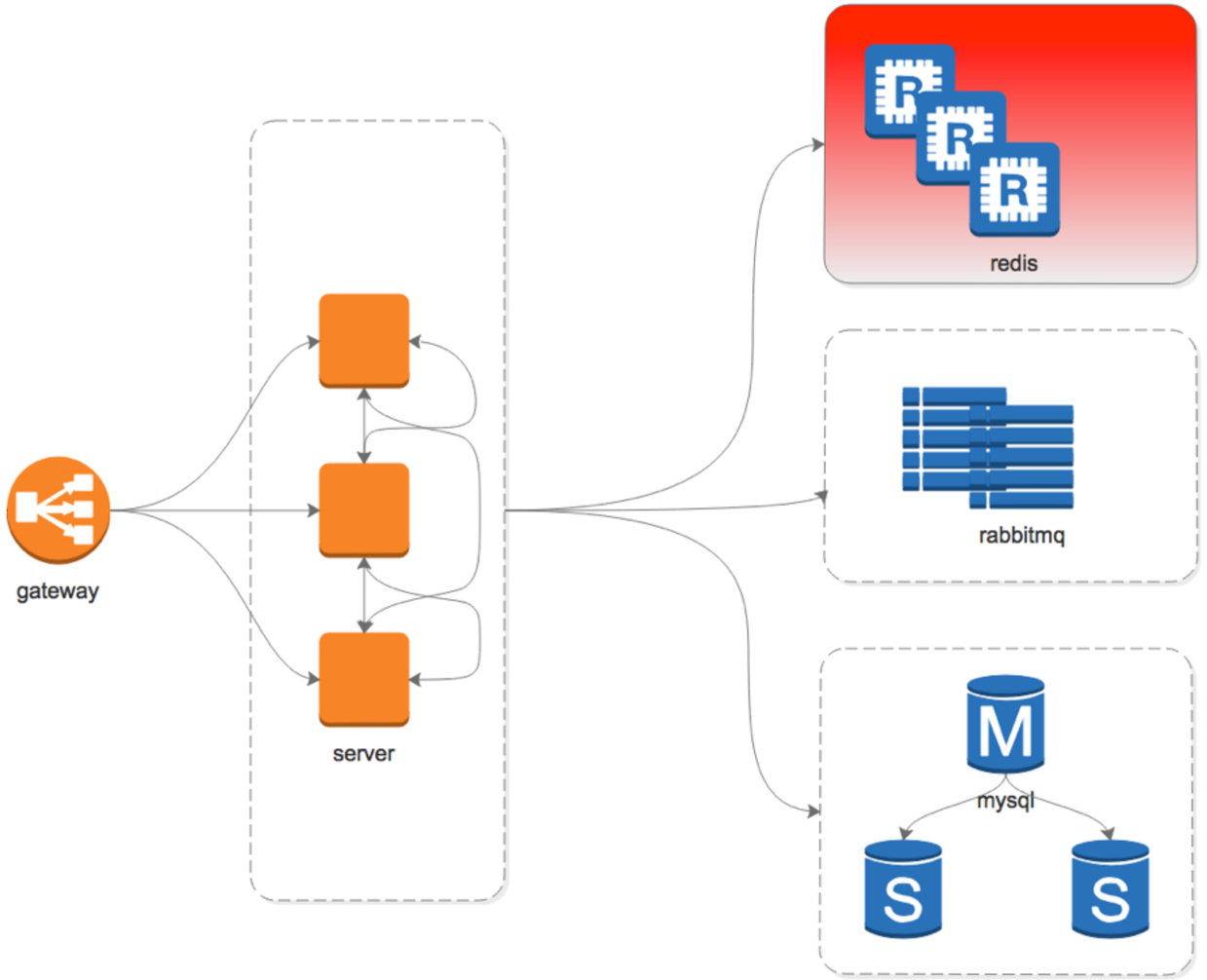
降级

过载


construct a HystrixCommand
1.
...or HystrixObservableCommand
.execute( )
.queue( )
.observe( )
.toObservable( )
2.
Then, choose one of the above methods to obtain the result(s) of the command. Note that they all ultimately depend on .toObservable( )
3.
available in cache?
no
4.
circuit-breaker open?
no
yes
short-circuit
5.
Semaphore / Thread pool rejected?
no
yes
reject
6.
construct( ) or run( )
7.
calculate circuit health
report metrics
yes; return cached response
8.
getFallback( ) or resumeWithFallback( )
execution fails?
yes
no
timeout?
yes
fallback successful?
no; failed or not implemented
yes; return fallback
9.
no; return resulting Observable
Legend:
invoke path
return path
observable:
items emitted
successful completion
error
Hystrix[Observable]Command method

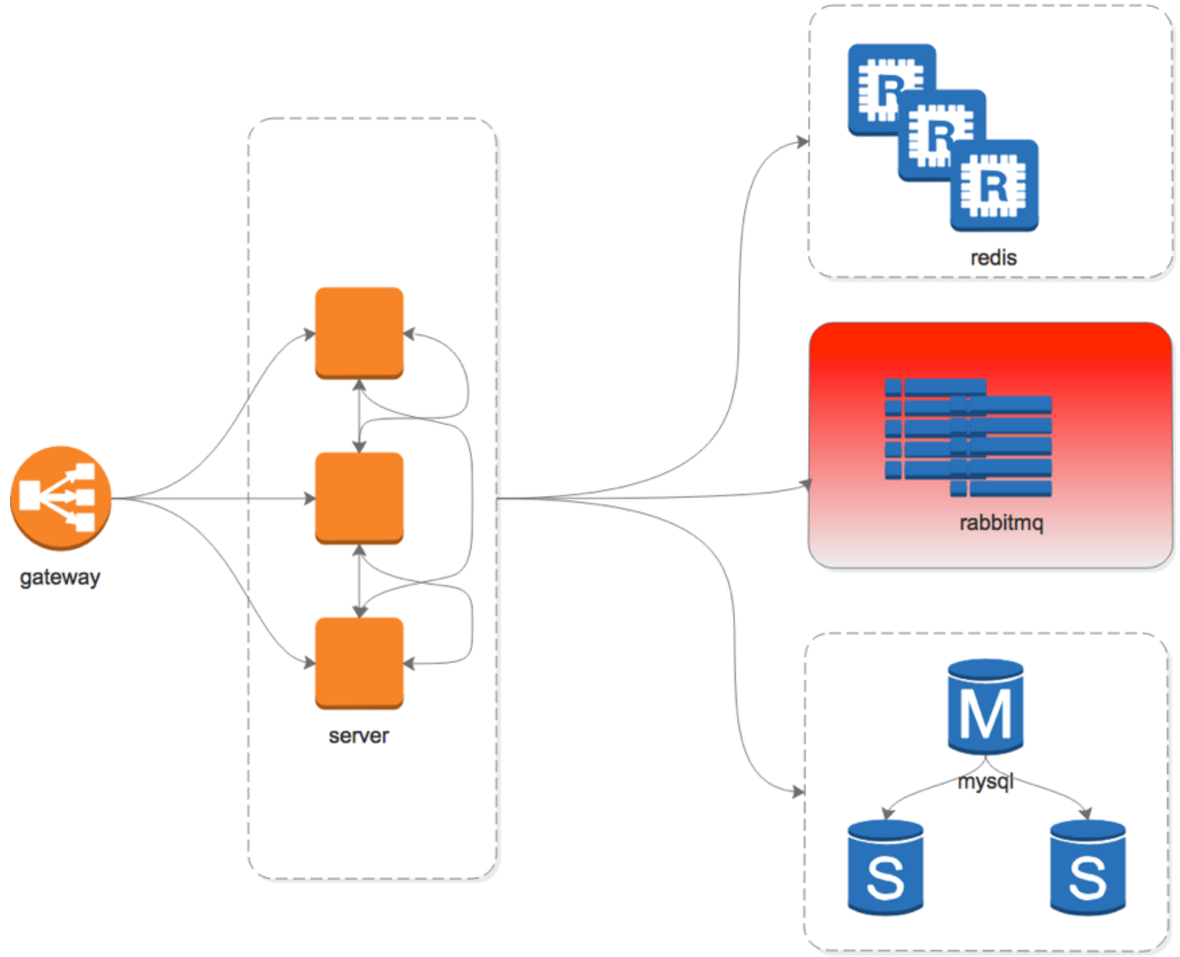
CACHE

Redis
Memcached



MESSAGEQUEUE

Kafka
RabbitMq



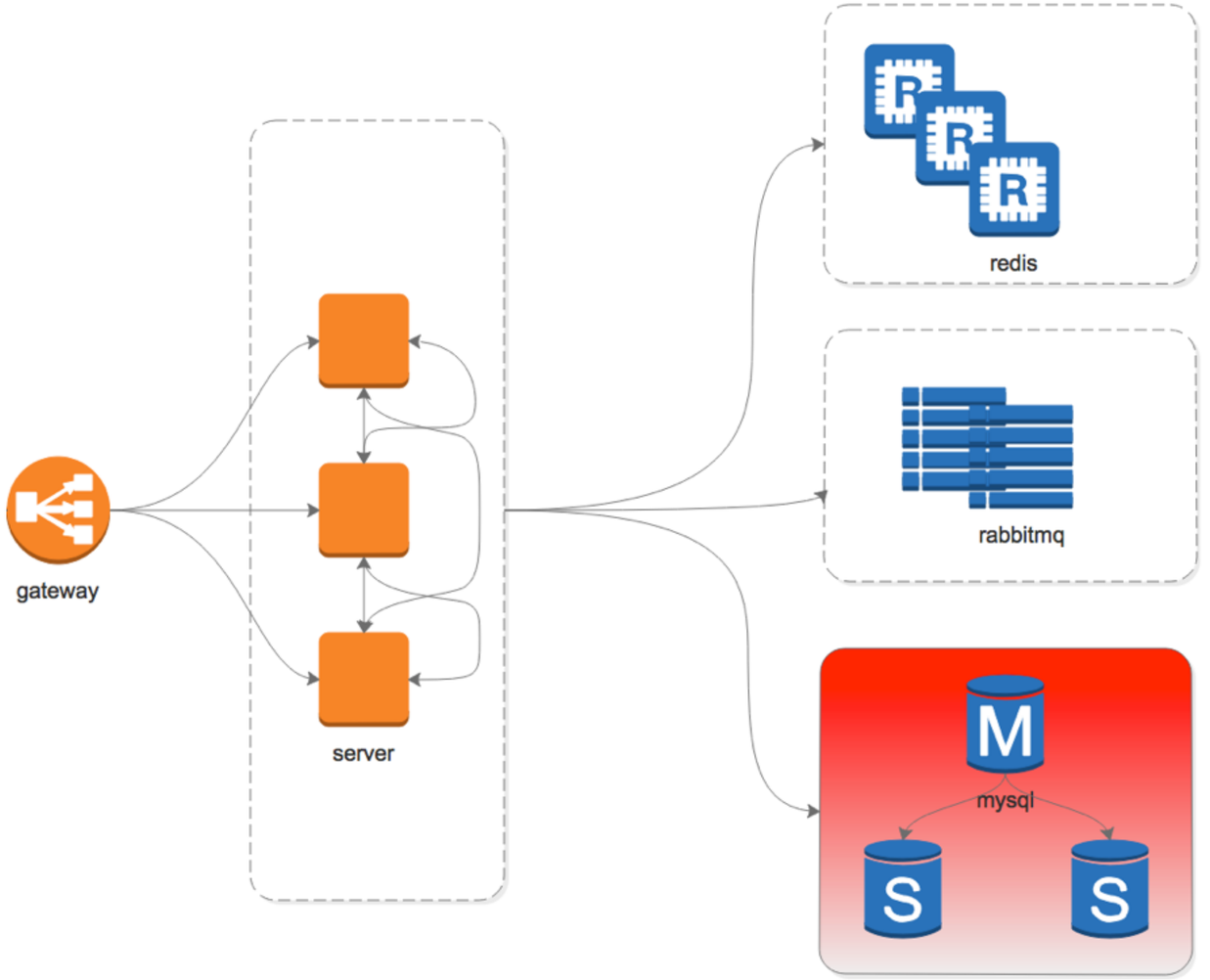
DATABASE

Mysql

高可用

读写分离

一致性



监控&发布

日志分析 elk接入
Metrics prometheus
应用监控 cat health endpoint 分布式定时任务调度(重试，幂等)
链路监控 zipkin
服务器监控 zabbix
无损发布

Q&A