**go-micro**

go-micro 微服务框架

go-kit 微服务框架

单体-> 多个业务为主的集群服务

传统框构->业务耦合高

传统框构进行更小粒度的切分-> 独立集群布置

Php-> swoole

Java ->sping 系统 spring boot +duubo

Java 天下第一，

Php才是最好的语言;

Go-micro

在现在容器化时代 => go更适合做容器

Go 编译之后可以直接 跑在ARM dockor iamge

Ele 他们之前用的java 去容器，服务器成本开销大

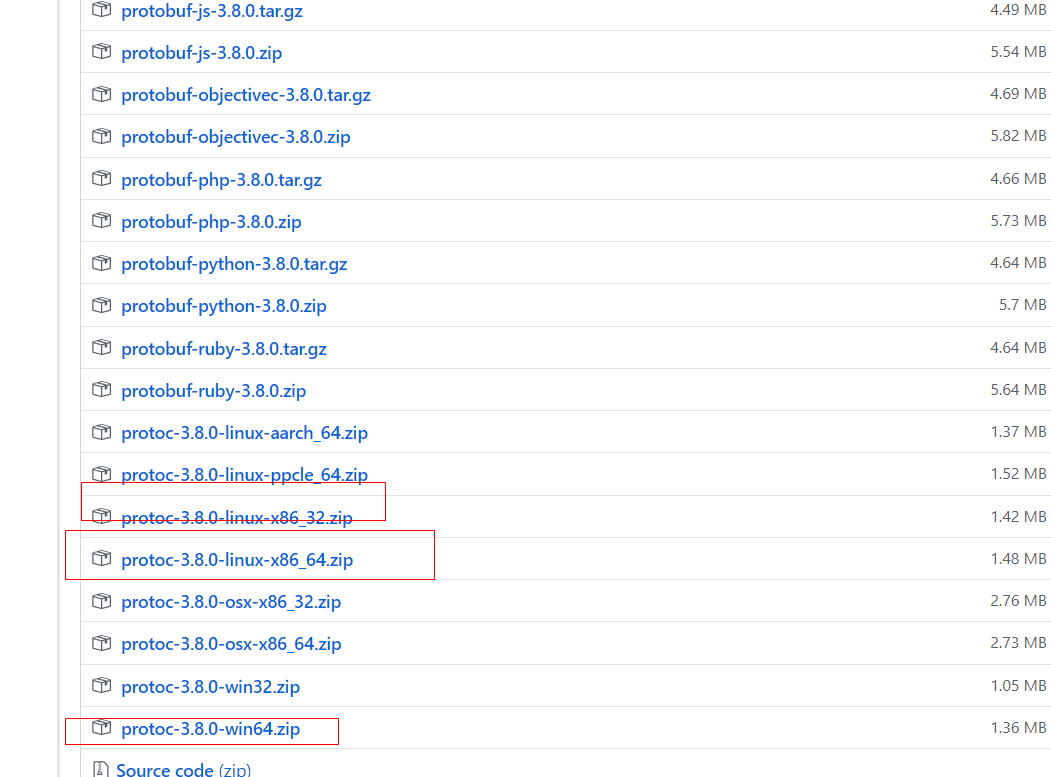
使用go做容器 ，服务器成本大概每年可以节约亿位数

Php做开发者，你只会php，那你就有点low,

GRPC 所需：

1. Protoc

Window



Go-mciro: hello word

1. 打开[https://github.com/micro/protoc-gen-micro](https://github.com/micro/protoc-gen-micro下载)
2. <https://github.com/google/protobuf>

找到自己系统对应的protoc程序文件下载，把文件目录加下环境变量

Protoc.exe

1. 验证cmd 执行protoc

拉取go get github.com/micro/protoc-ge

MDNS

什么是mdns ?

Dns ? UDP

Mdns 组播DNS

小明主机开启了Mdns Mdns-> 发送一条广播消息给局域网内所有开启MDNS服务 主机，

我来了，我ip和端口是多少

用来作服务注册，发现，查询

go.mod

replace (  
 github.com/golang/lint => golang.org/x/lint v0.0.0-20190409202823-959b441ac422  
 github.com/testcontainers/testcontainer-go => github.com/testcontainers/testcontainers-go v0.0.4  
)

如果出现以下error

build command-line-arguments: cannot load github.com/hashicorp/consul/api: ambiguous import: found github.com/hashicorp/consul/api in multiple modules:

github.com/hashicorp/consul v1.4.4 (C:\Users\vector\go\pkg\mod\github.com\hashicorp\consul@v1.4.4\api)

github.com/hashicorp/consul/api v1.1.0 (C:\Users\vector\go\pkg\mod\github.com\hashicorp\consul\api@v1.1.0)

解决方法如下

go get github.com/hashicorp/consul

1. Features
2. Service Discovery

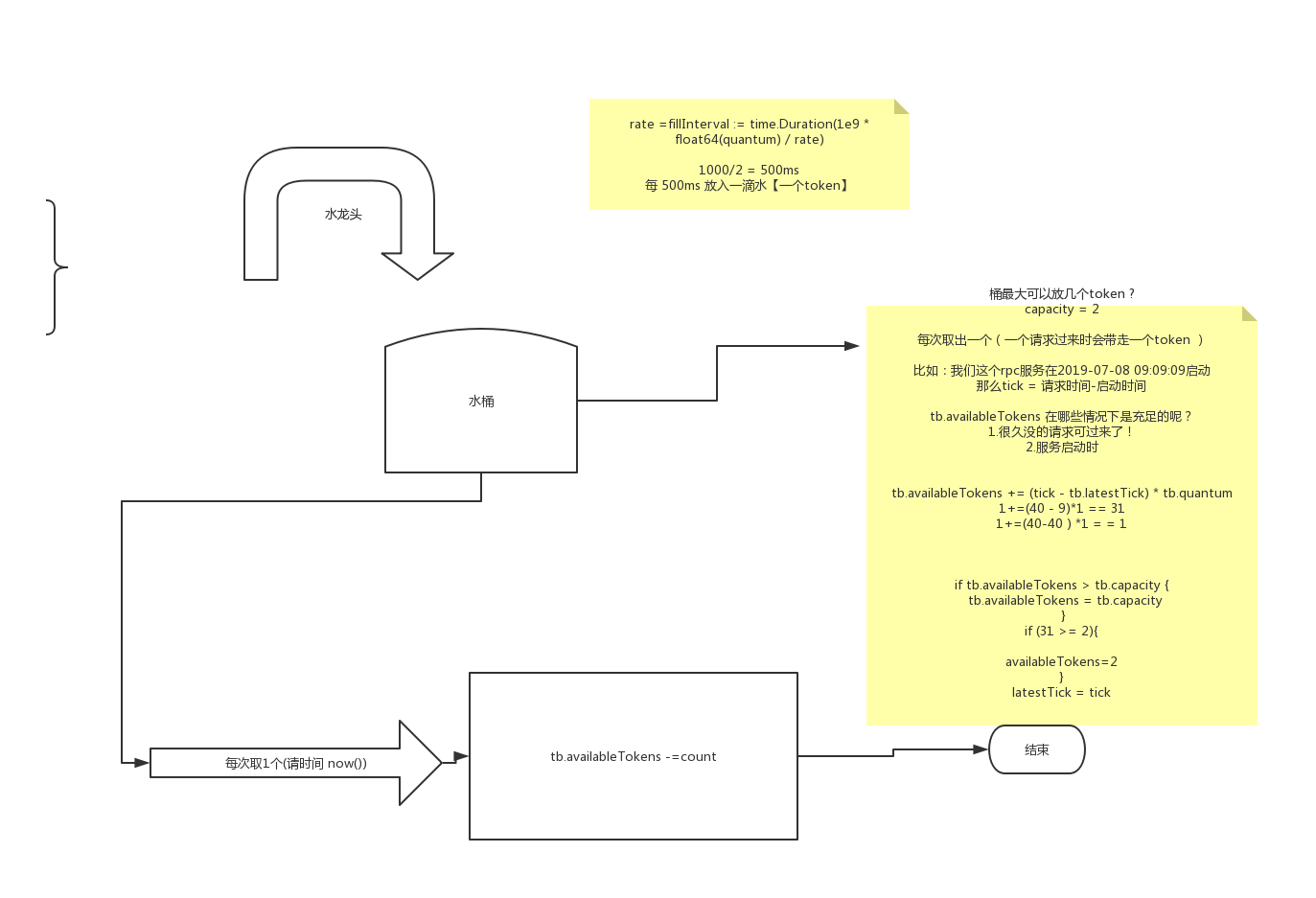
Dmns

Etcd

1. Load Balancing

Cousl etcd => k8s 默认使用

限流原图



**CentOS Docker 安装**

Docker支持以下的CentOS版本：

* CentOS 7 (64-bit)

**使用 yum 安装（CentOS 7下）**

Docker 要求 CentOS 系统的内核版本高于 3.10 ，查看本页面的前提条件来验证你的CentOS 版本是否支持 Docker 。

通过 **uname -r**命令查看你当前的内核版本

[root@runoob ~]# uname -r

https://www.runoob.com/wp-content/uploads/2016/05/docker08.png

**安装 Docker**

从 2017 年 3 月开始 docker 在原来的基础上分为两个分支版本: Docker CE 和 Docker EE。

Docker CE 即社区免费版，Docker EE 即企业版，强调安全，但需付费使用。

本文介绍 Docker CE 的安装使用。

移除旧的版本：

$ sudo yum remove docker \

docker-client \

docker-client-latest \

docker-common \

docker-latest \

docker-latest-logrotate \

docker-logrotate \

docker-selinux \

docker-engine-selinux \

docker-engine

安装一些必要的系统工具：

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

添加软件源信息：

sudo yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

更新 yum 缓存：

sudo yum makecache fast

安装 Docker-ce：

sudo yum -y install docker-ce

启动 Docker 后台服务

sudo systemctl start docker

其它系统见 <https://www.runoob.com/docker/centos-docker-install.html>

安装 dockers-composer

yum install docker-composer

编写 docker-compose.yml （随便找一个目录）

version: "3" #版本信息

services: #服务列表

redis: #定义一个redis服务

image: redis:4 # 镜像：镜像名：版本

ports: # 服务监听的端口信息映射关系

- 6379:6379

volumes: # 目录映射

- "/root/redis/conf:/usr/local/etc/redis"

- "/root/redis/data:/data"

command: # 运行命令

redis-server

mysql:

image: mysql:5.7

ports:

- 3306:3306

volumes:

- "/root/mysql/data:/var/lib/mysql"

- "/root/mysql/conf.d:/etc/mysql/conf.d"

environment:

- MYSQL\_ROOT\_PASSWORD=123456

rabbitmq:

image: rabbitmq:management-alpine

container\_name: rabbitmq\_server

ports:

- "15672:15672"

- "5672:5672"

environment:

- RABBITMQ\_DEFAULT\_USER=guest

- RABBITMQ\_DEFAULT\_PASS=guest

## 2. docker-compose常用命令

docker-compose dup [options] [SERVICE...]

该命令十分强大，它将尝试自动完成包括构建镜像，（重新）创建服务，启动服务，并关联服务相关容器的一系列操作。  
链接的服务都将会被自动启动，除非已经处于运行状态。

docker-compose down

此命令将会停止 up 命令所启动的容器，并移除网络

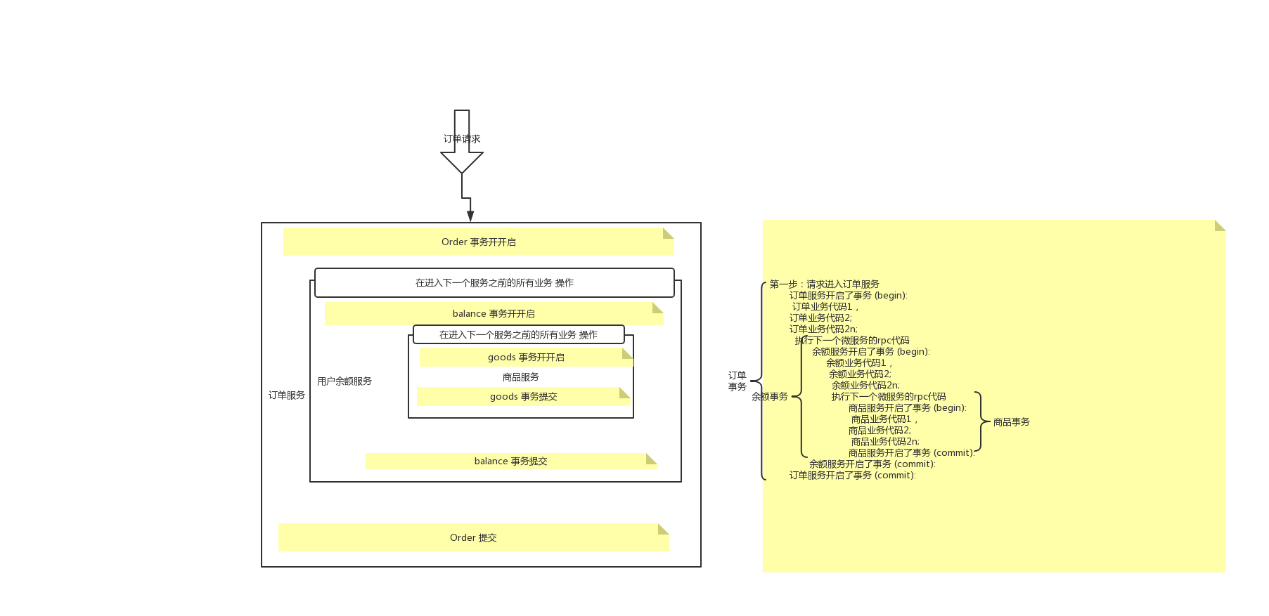
docker-compose restart [options] [SERVICE...]

重启项目中的服务。

# 访问管理界面

访问管理界面的地址就是 http://[宿主机IP]:15672，可以使用默认的账户登录，用户名和密码都guest;

利用代码实现事务嵌套



<https://github.com/yakaa/integral-mall> 把对应的common目录复制到application