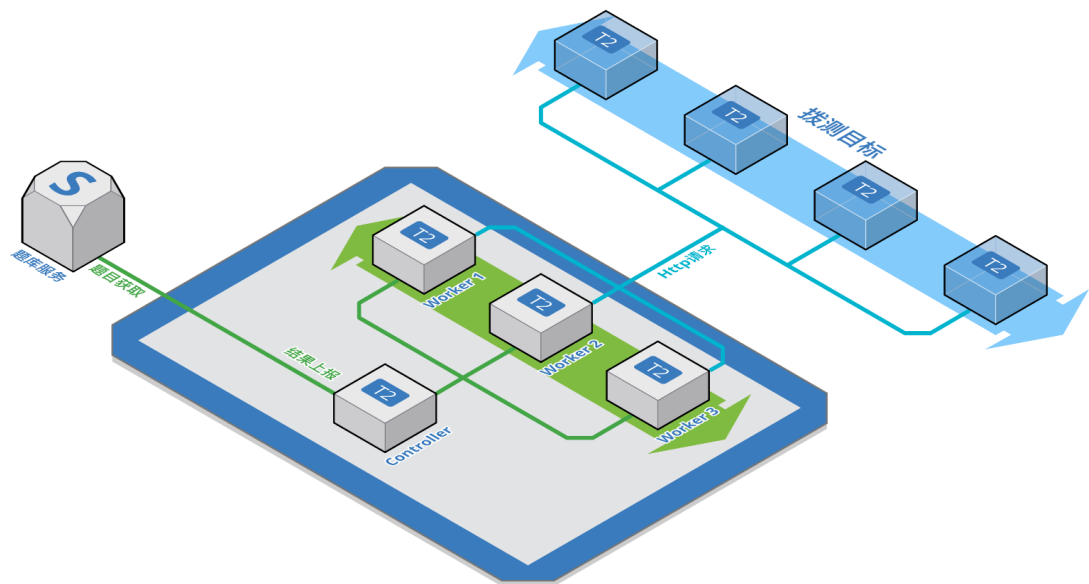


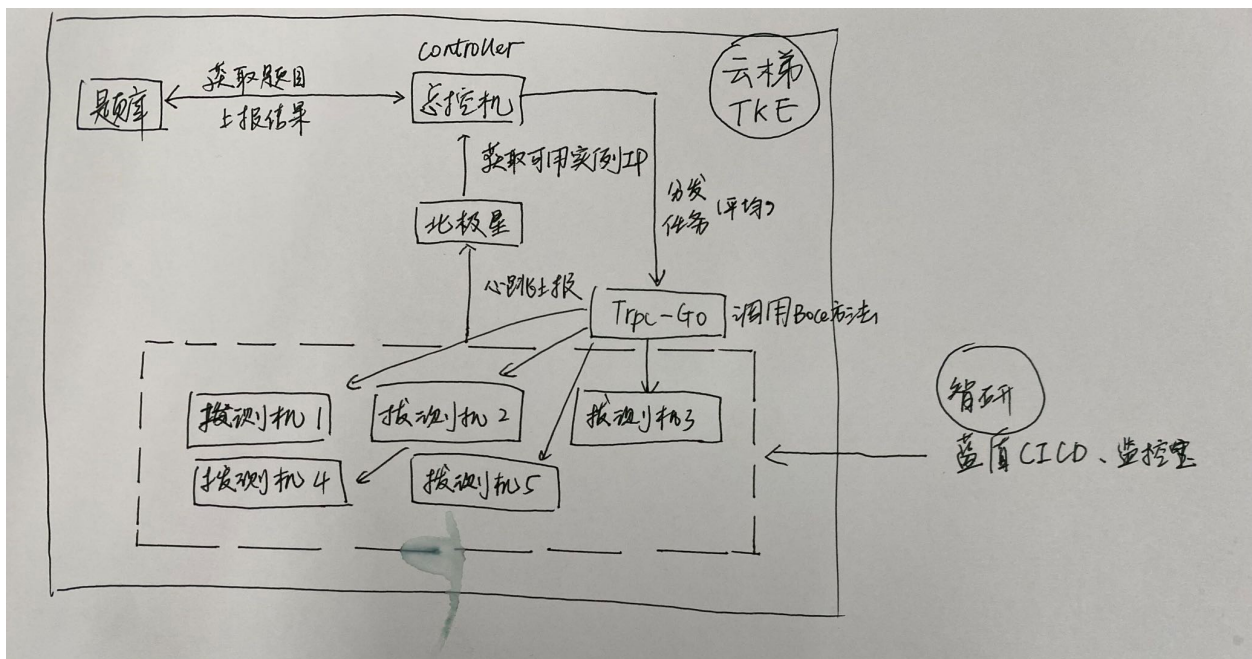
showerli 拨测系统

一、系统介绍



系统由1台总控机 (Controller) 和多台拨测机 (Worker) 组成。总控机从对应的题库服务器拉取拨测任务并下发给拨测机进行任务拨测，拨测机接收到任务后向对应的目标服务器发起 Http 请求，将响应结果上报回总控机，总控机汇总后将拨测结果统一上报回题库服务器进行结果裁定。

1.1 系统框架



1.2 系统标准

考察内容	考察标准	分值	完成情况
数据准确率	响应结果与题库中的结果一致	20	100%准确率
基础功能完整	3m内完成1W拨测量，准确率99%以上	20	1w请求平均1.3秒
极限负载	高于准确率及格线的单次最大任务数	15	14万
系统可用性	负载均衡、横向扩容、宕机自愈、自我负载管控	15	接入北极星、云梯TKE服务
系统完备性	监控管理、告警管理、OSS、自动上下架	15	TKE日志汇、智研监控宝
框架应用	Trpc	5	使用Trpc-go框架
公共组件	tke、蓝盾、智研/灵动	10	蓝盾流水线showerli-worker
其他	以上仅做参考，同学们有其他创意和想法可自行添加，系统完善性作为加分项考虑	无上限	工蜂、腾讯软件源仓库

二、使用展示

在远程终端使用：./controller amount命令，其中**第二个参数为请求数量（100的倍数）**

```
[root@showerli-service-controller-5cf4c88dcc-7dvfh /controller]# ./controller 10000
正在向题库获取题目，请求数量为： 10000
*****获取题库成功*****
*****北极星获取实例IP成功*****
worker: 9.165.220.97:8080 拨测任务执行完毕
worker: 9.165.177.39:8080 拨测任务执行完毕
worker: 9.166.209.132:8080 拨测任务执行完毕
worker: 9.166.242.247:8080 拨测任务执行完毕
worker: 9.166.144.56:8080 拨测任务执行完毕
*****正在将结果上报给题库*****
{"code":200,"msg":"更新成功","data":{"rate":100,"cost_time":"1159 ms"}}
[root@showerli-service-controller-5cf4c88dcc-7dvfh /controller]# ./controller 10000
正在向题库获取题目，请求数量为： 10000
*****获取题库成功*****
*****北极星获取实例IP成功*****
worker: 9.166.144.56:8080 拨测任务执行完毕
worker: 9.166.209.132:8080 拨测任务执行完毕
worker: 9.166.242.247:8080 拨测任务执行完毕
worker: 9.165.177.39:8080 拨测任务执行完毕
worker: 9.165.220.97:8080 拨测任务执行完毕
*****正在将结果上报给题库*****
{"code":200,"msg":"更新成功","data":{"rate":100,"cost_time":"1342 ms"}}
```

最终速度、最大请求数统计：

rtx	单次任务最大请求数	单次任务最快处理速度 (条/秒, 与最大请求数无关)
ediewu	220,000	6,018
jodehe	200,000	8,010
showerli	140,000	16,779
wadeyli	120,000	7,794
ricardohan	100,000	49,751
zevycai	100,000	42,194
shufengli	100,000	16,639
sakurapeng	100,000	12,887
elandluo	100,000	11,891
yingxzhang	100,000	9,307

< 1 2 > 20 条/页

三、代码说明

3.1 controller

```

1 func main() {
2     task := Gettask() // task为Response结构体
3     fmt.Println("*****获取题库成功*****")
4     var IPList = Polaris() //从北极星获取实例IP
5     fmt.Println("*****北极星获取实例IP成功*****")
6     var distribute_task WorkerTask
7     sum, _ := strconv.Atoi(amount[1])
8     unit := sum / len(task.URLList) / len(IPList) // 已知url的数量固定
    为20个, amount = 20 * add
9     //fmt.Println("一共有",len(task.AddrList),"个add")
10    //fmt.Println("一共有",len(task.URLList),"url")
11    var output_son = make(chan result_list)
12    //var wg = sizedwaitgroup.New(100)
13    for i := 0; i < len(IPList); i++){
14        distribute_task.Url_list = task.URLList //每个worke要遍历所有url
15        distribute_task.AddrList = task.AddrList[ i*unit : (i+1)*unit ]
16        //wg.Add()
17        go workerBoce(distribute_task, IPList[i], output_son)
18        //defer wg.Done()
19    }
20    //wg.Wait()
21    var output_father result_list
22    i := 0
23    for s := range output_son {

```

```

25     output_father = append(output_father,s...)
26     if i >= len(IPlist) -1 {
27         close(output_son)
28     }
29     i++
30 }
31 shangbao(task.TaskUUID, "showerli", output_father) //上报函数将结果
    上报给题库
32 }

```

3.2 worker

```

1 // Boce方法
2 func (s *greeterServiceImpl) Boce(ctx context.Context, req
    *pb.HelloRequest, rsp *pb.HelloReply) error {
3     question := req.Msg //获取入参值，是一个字符串数组
4     var distribute_task WorkerTask
5     _ = json.Unmarshal([]byte(question), &distribute_task)
6     //fmt.Println(distribute_task)
7     var t = make(chan record)
8     addle := len(distribute_task.AddrList)
9     urlle := len(distribute_task.Url_list)
10    for i := 0; i < addle; i++ {
11        for k := 0; k < urlle; k++ {
12            if distribute_task.Url_list[k].Method == "GET" {
13                go GEThttp(ctx, distribute_task.AddrList[i],
                    distribute_task.Url_list[k].URL, t)
14            } else {
15                go HEADhttp(ctx, distribute_task.AddrList[i],
                    distribute_task.Url_list[k].URL, t)
16            }
17        }
18    }
19
20    i := 0
21    var reslut []record // worker拨测结果
22    for s := range t {
23        reslut = append(reslut, s)
24        if i >= addle * urlle -1 {
25            close(t)
26        }

```

```

27         i++
28     }
29
30     reslutList, err := json.Marshal(reslut)
31     if err != nil {
32         return err
33     }
34     rsp.Msg = string(reslutList)
35     //fmt.Println(string(reslutList))
36     return nil
37 }

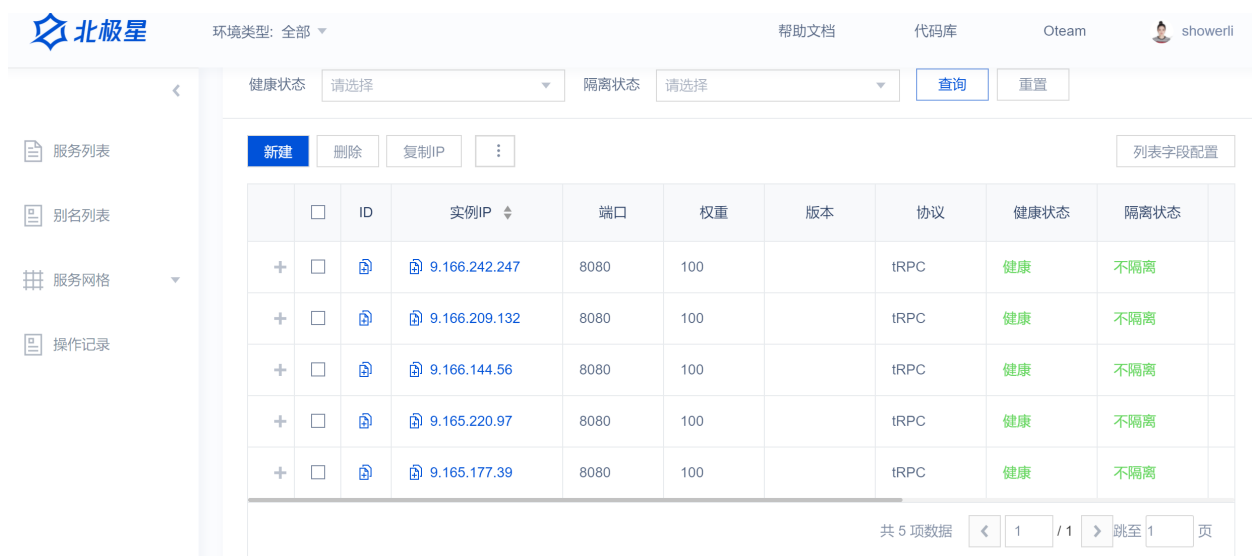
```

四、系统可用性与完备性

4.1 负载均衡

北极星服务用于解决远程调用（RPC）的服务注册发现、动态路由、负载均衡和容错问题，有四种使用方式。本系统采用北极星API，支持HTTP/gRPC/tRPC协议，应用程序可以直接调用服务端服务注册和发现接口。

showerli-controller在北极星注册了服务并绑定服务实例，北极星服务链接：[北极星\(oa.com\)](http://oa.com)



The screenshot shows the Polarix (北极星) web interface. The top navigation bar includes the logo, environment type (全部), help documentation, code repository, and user information (Oteam, showerli). The main content area displays a table of service instances with columns for ID, instance IP, port, weight, version, protocol, health status, and isolation status. The table shows 5 instances, all with a health status of '健康' (Healthy) and '不隔离' (Not Isolated). The bottom of the interface shows pagination information: '共 5 项数据' (Total 5 items) and '1 / 1' pages.

ID	实例IP	端口	权重	版本	协议	健康状态	隔离状态
9.166.242.247	8080	100		tRPC	健康	不隔离	
9.166.209.132	8080	100		tRPC	健康	不隔离	
9.166.144.56	8080	100		tRPC	健康	不隔离	
9.165.220.97	8080	100		tRPC	健康	不隔离	
9.165.177.39	8080	100		tRPC	健康	不隔离	

polaris()用于从北极星获取可用实例。由于北极星API调取速度较慢，每一个IP获取需要0.2左右，多worker情况下会拖慢时间。这里polaris()这里直接返回包含所有可用实例IP的数组。

```

1 func Polaris() []string {
2     initArgs()

```

```

3     consumer, err := api.NewConsumerAPI()
4     if err != nil {
5         log.Fatalln(err)
6     }
7     defer consumer.Destroy()
8     deadline := time.Now().Add(time.Duration(seconds) * time.Second)
9     for {
10        if time.Now().After(deadline) {
11            //break
12        }
13        var flowId uint64
14        instance_req := &api.GetAllInstancesRequest{}
15        instance_req.FlowID = atomic.AddUint64(&flowId, 1)
16        instance_req.Namespace = namespace
17        instance_req.Service = service
18        getInstResp, err := consumer.GetAllInstances(instance_req)
19        if err != nil {
20            log.Fatalln(err)
21        }
22        targetInstanceIp := make([]string, 0)
23        for _, target := range getInstResp.Instances {
24            targetInstanceIp = append(targetInstanceIp,
target.GetHost()+":"+strconv.FormatUint(uint64(target.GetPort()), 10))
25        }
26        // 最终返回由实例IP组成的的字符串数组
27        return targetInstanceIp
28    }
29 }

```

4.2 横向扩容、自我负载管控

showerli-controller和showerli-worker采用云梯TKE的Deployment工作负载类型去部署任务，去实现容器、负载的管理，调节、操控。由于Deployment是无状态的部署Pod，使用浮动IP时会默认随时回收。

showerli-worker链接：[云梯-助力业务上云 \(woa.com\)](https://woa.com)

showerli-controller链接：[云梯-助力业务上云 \(woa.com\)](https://woa.com)

名称

worker

最长63个字符，只能包含小写字母、数字及分隔符("-",)，且不能以分隔符开头或结尾

镜像

mirrors.tencent.com/showerli_

镜像版本 (Tag)

worker

同时支持公共镜像与私有镜像，详细介绍请参考[指引](#)。

CPU/内存限制

CPU限制

request 3 - limit 5 核

内存限制

request 1 - limit 3 GiB

Request用于预分配资源,当集群中的节点没有request所要求的资源数量时,容器会创建失败。
Limit用于设置容器使用资源的最大上限,避免异常情况下节点资源消耗过多。

环境变量

[新增变量](#)

变量名可以包含大小写字母、数字、中划线、下划线和英文句号，并且只能以字母开头

[显示高级设置](#)

注意：Workload创建完成后，容器的配置信息可以通过更新YAML的方式进行修改

在创建工作负载时候进行配置，当内存或者CPU负载过高时，工作负载会自动扩容，以提升负载能力，当负载降低时，会自动缩容，降低成本：

实例数量

☐ 手动调节 直接设定实例数量

☒ 自动调节 满足任一设定条件，则自动调节实例（pod）数目

触发策略

CPU利用率

70

%

×

内存利用率

70

%

×

[新增策略](#)

实例范围

4 ~ 8

在设定的实例范围内自动调节，不会超出该设定范围

满足设定的任意一个条件，实例（pod）数量会自动调节：

<input type="checkbox"/> 实例名称	状态	镜像	实例所在节点IP	实例IP	运行时间	操作
<input type="checkbox"/> showerli-service-7bbf	Running	mirrors.tencent.com/showerli_booce/booce:worker	11.167.45.83	9.165.220.97	15 小时	销毁实例 远程登录 更多
<input type="checkbox"/> showerli-service-7bbf	Running	mirrors.tencent.com/showerli_booce/booce:worker	11.167.45.153	9.165.177.39	15 小时	销毁实例 远程登录 更多
<input type="checkbox"/> showerli-service-7bbf	Running	mirrors.tencent.com/showerli_booce/booce:worker	11.134.129.139	9.166.242.247	15 小时	销毁实例 远程登录 更多
<input type="checkbox"/> showerli-service-7bbf	Running	mirrors.tencent.com/showerli_booce/booce:worker	11.161.29.102	9.166.144.56	15 小时	销毁实例 远程登录 更多
<input type="checkbox"/> showerli-service-7bbf	Running	mirrors.tencent.com/showerli_booce/booce:worker	11.161.31.99	9.166.209.132	15 小时	销毁实例 远程登录 更多

第 1 页 20 条 / 页 < >

4.3 宕机自愈

TKE中的容器实例出现程序有BUG、容器run failed、dockerfile拉取的基础镜像出错等问题时，容器会挂掉，进程会终止，这时候TKE会自动重新拉取镜像，然后尝试重新拉起pod。同时，当我们的pod挂掉时北极星的实例队列就会剔除该实例，当该容器能够正常工作时，又会加入到队列里来。

首次出现时间	最后出现时间	级别	资源类型	资源名称	内容	详细描述	出现次数
2021-08-04 23:10:07	2021-08-05 10:58:18	Warning	Pod	showerli-worker-78d5fc7d8b-gf9r2.1698230dbdf6cd89f0	BackOff	Back-off restarting failed container	3264
2021-08-04 23:08:14	2021-08-05 10:23:22	Normal	Pod	showerli-worker-78d5fc7d8b-gf9r2.169822f37b5b571c0f	Pulling	Pulling image "mirrors.tencent.com/sho..."	137

4.4 CI/CD

showerli拨测系统使用智研交付流+蓝盾流水线实现CI/CD。首先蓝盾流水线监听Git push事件，自动启动流水线：

>

Git事件触发

引用变量

插件: [帮助文档](#)

版本:

Git事件触发

重选

1.latest

代码库: *

☒ 按代码库选择

☐ 按代码库别名输入

showerli/HttpBoce

事件类型:

☒ Commit Push Hook

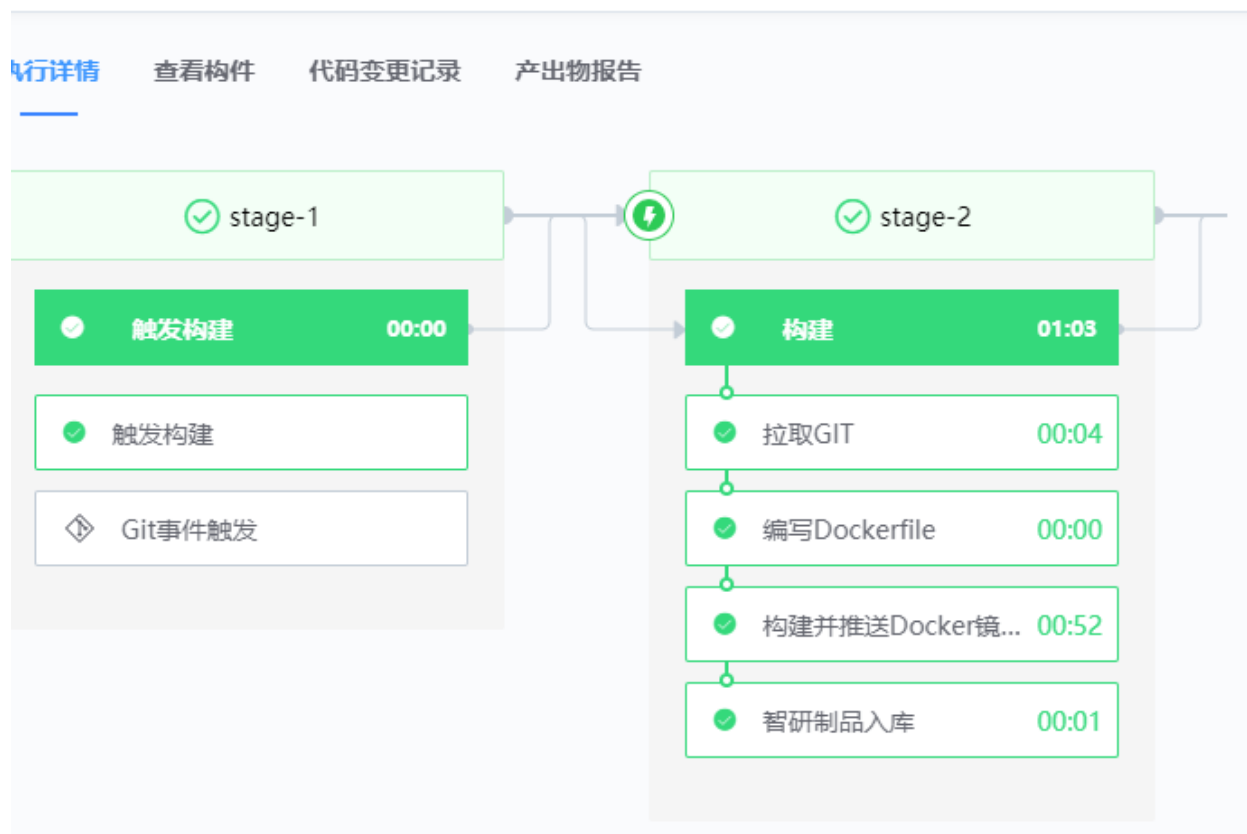
☐ Tag Push Hook

☐ Merge Request Hook

☐ Merge Request Accept Hook

☐ Code Review Hook

然后拉取代码，构建镜像，并且打包存入智研制品库。[showerli-worker流水线链接: 蓝盾DevOps平台 \(oa.com\)](#)



在智研交付流-部署中可以把打包好的镜像一键部署到云梯TKE，实现CI/CD：

交付流 / 部署 / 部署配置

部署配置 部署历史 版本分布 实例列表

应用名称 showerli-worker 环境名称 请选择环境 查询

新增部署配置

部署配置	应用名称	环境名称	环境状态	最近部署状态	最近部署时间	最近操作人	操作
showerli-worker部署prod	showerli-worker	默认生产环境	允许变更				部署新实例 发布 部署历史 管理实例

共 1 项数据

4.5 日志与监控

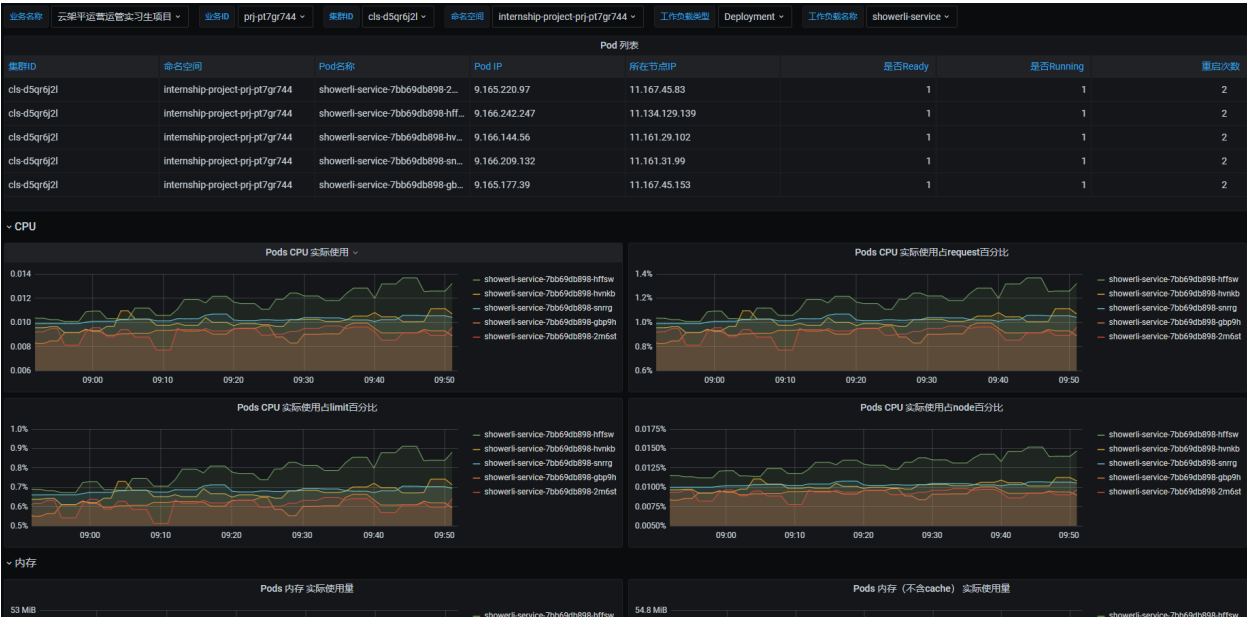
showerli拨测系统可以一键查看pod情况，做到了实时监控：

监控

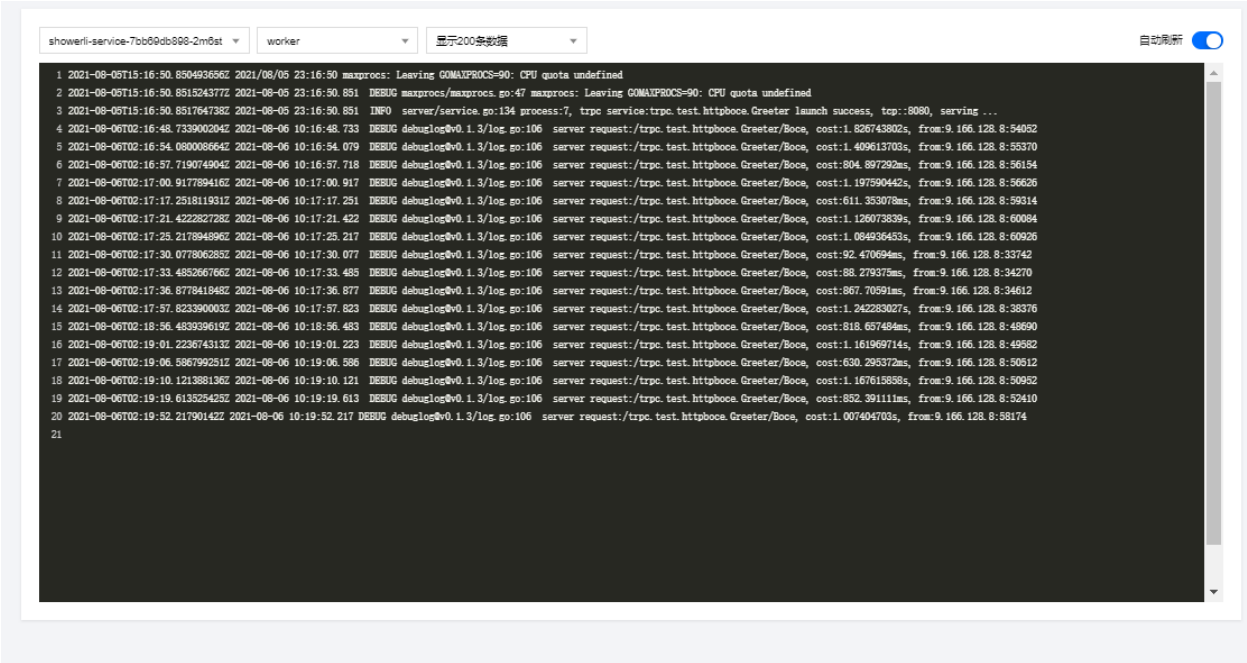
多个关键字用空格分隔，多个过滤器用回车分隔

实例名称	状态	镜像	实例所在节点IP	实例IP	运行时间①	操作
showerli-service-7bbf	Running	mirrors.tencent.com/showerli_boce/boce-worker	11.107.45.83	9.105.220.97	15 小时	销毁重建 远程登录 更多
showerli-service-7bbf	Running	mirrors.tencent.com/showerli_boce/boce-worker	11.107.45.153	9.105.177.39	15 小时	销毁重建 远程登录 更多
showerli-service-7bbf	Running	mirrors.tencent.com/showerli_boce/boce-worker	11.134.129.139	9.105.242.247	15 小时	销毁重建 远程登录 更多
showerli-service-7bbf	Running	mirrors.tencent.com/showerli_boce/boce-worker	11.101.29.102	9.105.144.55	15 小时	销毁重建 远程登录 更多
showerli-service-7bbf	Running	mirrors.tencent.com/showerli_boce/boce-worker	11.101.31.99	9.105.209.132	15 小时	销毁重建 远程登录 更多

第1页 20 条 / 页



可以查看各pod的日志:



5. 问题与反思

5.1 网络环境隔离

5.2 反思