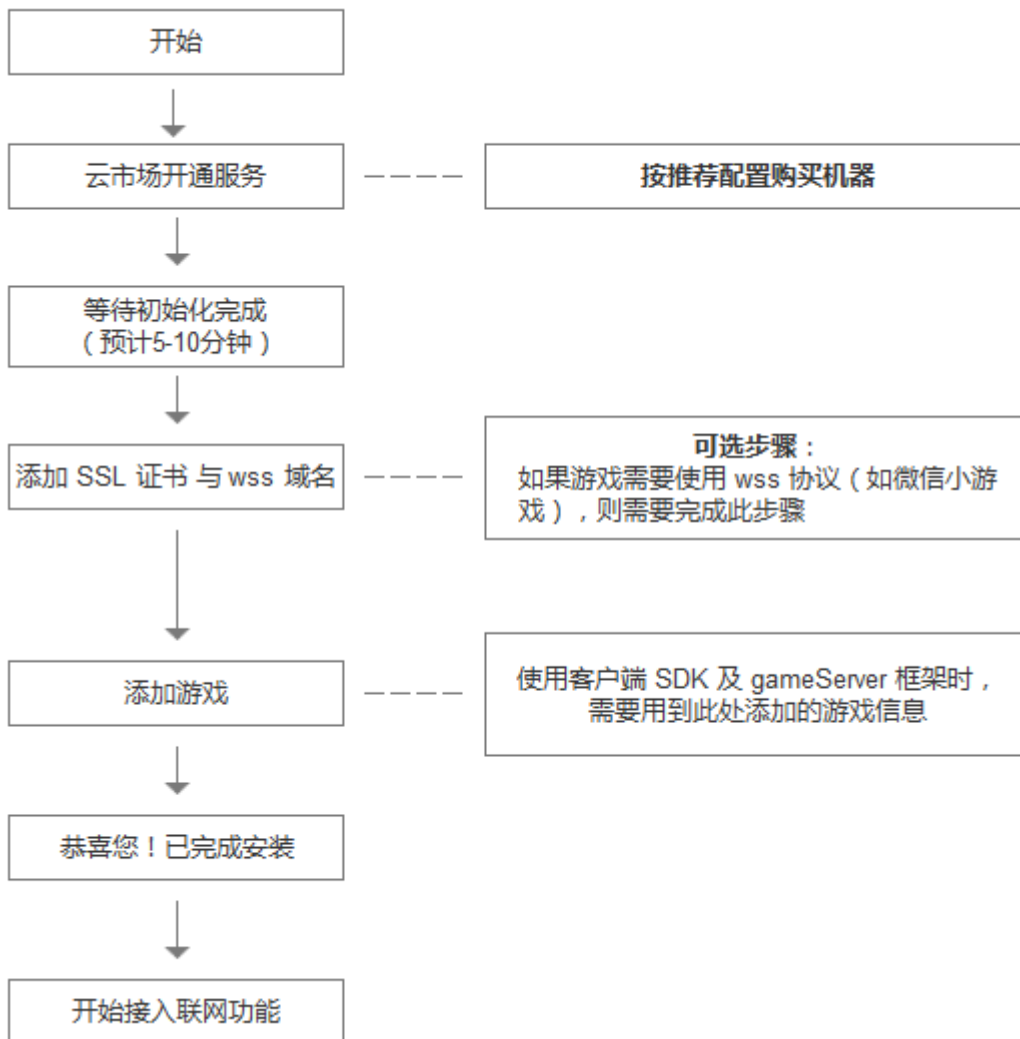


# Matchvs 游戏云镜像版教程

## 安装步骤概览



## 开通服务

在阿里云市场搜索 Matchvs 或者 Matchvs 镜像版，如图：



选择立即开通服务，按照如下推荐配置进行服务器购买：

## 推荐配置表

可以根据  $\text{单房间人数} * \text{发包频率} * \text{包长} * \text{CCU}$  的计算结果购买所需带宽，  
如果带宽很高，推荐使用阿里云的共享带宽

<b>配置 A</b> ecs.c5.xlarge 4核	<b>配置 B</b> ecs.c5.2xlarge 8核	<b>配置 C</b> ecs.c5.3xlarge 12核	<b>配置 D</b> ecs.c5.4xlarge 16核
<b>配置 E</b> ecs.c5.6xlarge 24核	<b>配置 F</b> ecs.c5.8xlarge 32核	<b>配置 G</b> 推荐使用SaaS或 多机器独立部署	



提示

\* 下面表格中的 A = 配置A / B = 配置B / C = 配置C / D = 配置D / E = 配置E / F = 配置F / G = 推荐方案

\* CCU 表示同时在线人数

\* 单房间最大人数支持20人以上；推荐配置为保障可用性会比实际需要偏高。  
通过配合GameServer的使用和降低转发消息数目可以提升同时在线并发能力。

单房间人数 (人)	单人发包频率 (个/秒)	CCU峰值	CCU峰值	CCU峰值	CCU峰值	CCU峰值	CCU峰值
		1-100 约 2000 DAU 约 4万 MAU	100-500 约 1万 DAU 约 20万 MAU	500-1000 约 2万 DAU 约 40万 MAU	1000-2000 约 4万 DAU 约 80万 MAU	2000-4000 约 8万 DAU 约 160万 MAU	4000-8000 约 16万 DAU 约 320 MAU
2	10	A	A	A	A	B	D
	20	A	A	A	B	C	E
	30	A	A	A	B	D	E
	60	A	A	B	C	E	G
6	10	A	A	A	B	D	F
	20	A	A	B	C	E	G
	30	A	A	B	C	E	G
	60	A	A	D	E	G	G
10	10	A	A	A	B	D	F
	20	A	A	B	D	E	G
	30	A	A	B	D	F	G
	60	A	B	D	F	G	G
20	10	A	A	B	D	E	G
	20	A	B	C	E	F	G
	30	A	B	D	E	G	G
	60	A	C	E	G	G	G

- Matchvs 推荐您使用服务器的配置不小于 4 vCPU 8GiB 内存 计算优化型的服务器，数据盘没有强制要求。

架构: **x86 计算** 异构计算 GPU / FPGA 弹性裸金属服务器 (神龙)

分类: 通用型 **计算型** 内存型 大数据型 本地 SSD 高主频型 入门级(共享)

规格族	实例规格	vCPU	内存	处理器型号	处理器主频	内网带宽	内网收发包	规格参考价格	
<input type="radio"/>	计算型 c5	ecs.c5.large	2 vCPU	4 GiB	Intel Xeon(SkyLake) Platinum 8163	2.5 GHz	1 Gbps	30 万 PPS	0.62 元/时
<input checked="" type="radio"/>	计算型 c5	ecs.c5.xlarge	4 vCPU	8 GiB	Intel Xeon(SkyLake) Platinum 8163	2.5 GHz	1.5 Gbps	50 万 PPS	1.24 元/时
<input type="radio"/>	计算型 c5	ecs.c5.2xlarge	8 vCPU	16 GiB	Intel Xeon(SkyLake) Platinum 8163	2.5 GHz	2.5 Gbps	80 万 PPS	2.49 元/时
<input type="radio"/>	计算型 c5	ecs.c5.3xlarge	12 vCPU	24 GiB	Intel Xeon(SkyLake) Platinum 8163	2.5 GHz	4 Gbps	90 万 PPS	3.73 元/时
<input type="radio"/>	密集计算型 ic5	ecs.ic5.large	2 vCPU	2 GiB	Intel Xeon(SkyLake) Platinum 8163	2.5GHz	1 Gbps	30 万 PPS	0.59 元/时
<input type="radio"/>	密集计算型 ic5	ecs.ic5.xlarge	4 vCPU	4 GiB	Intel Xeon(SkyLake) Platinum 8163	2.5GHz	1.5 Gbps	50 万 PPS	1.18 元/时
<input type="radio"/>	密集计算型 ic5	ecs.ic5.2xlarge	8 vCPU	8 GiB	Intel Xeon(SkyLake) Platinum 8163	2.5GHz	2.5 Gbps	80 万 PPS	2.36 元/时
<input type="radio"/>	密集计算型 ic5	ecs.ic5.3xlarge	12 vCPU	12 GiB	Intel Xeon(SkyLake) Platinum 8163	2.5GHz	4 Gbps	90 万 PPS	3.54 元/时

当前选择实例: ecs.c5.xlarge (4 vCPU 8 GiB, 计算型 c5)

购买实例数量: 1 台 已开通 0 vCPU, 还可开通 600 vCPU。当前所选实例规格为 4 vCPU, 最多还可开通 150 台 ECS

镜像: **系统镜像** 自定义镜像 共享镜像 **镜像市场**

当前选择的镜像: Matchvs 镜像版 v1.0.2

[重新选择镜像](#)

- 网络要求，**不小于10Mbps**，主要流量来自于对战通信数据。请根据游戏对战通信情况预估，并关注带宽用量酌情调整
- 阿里云安全组要求，开通**http(80) https(443) gateway(7001) hotel(9330)**

分配公网IP地址 系统会分配公网IP，也可采用更加灵活的弹性公网IP方案，配置并 [绑定弹性公网IP地址](#)

按使用流量  按固定带宽 带宽费用合并在ECS实例中收取

1M 25M 50M 75M 100M  Mbps

阿里云免费提供最高5Gbps的恶意流量攻击防护，[了解更多](#) | [提升防护能力](#)

---

**重新选择安全组** 安全组类似防火墙功能，用于设置网络访问控制，您也可以到管理控制台 [新建安全组](#) [教我选择](#)

所选安全组：测试服务器 / sg-2zeewci6w1epe7aw9vp7 (已有2个实例+辅助网卡，还可以加入1998个实例+辅助网卡)

请确保此安全组开放包含22 (Linux) 或者3389 (Windows) 端口，否则无法远程登录ECS。您可以进入ECS控制台设置。[前往设置](#)

---

eth0: 主网卡 交换机: [默认]vsw-2ze2mgsb0bbeoh2cvzr8  自动分配IP地址  随实例释放

**+** 增加弹性网卡 您还可增加1块

## 等待初始化

开通服务，并启动机器后，镜像会对机器进行初始化操作。在阿里云控制台等到ECS服务器状态显示为运行中：

实例ID/名称	标签	监控	可用区	IP地址	状态	网络类型	配置	付费方式	操作
i-2ze0wi launch-advisor-2...			华北2 可用区 E	172.17.187.20(私有)	<span style="border: 1px solid red; border-radius: 50%; padding: 2px;">运行中</span>	专有网络	ecs.g4gce (I/O优化)	2018年10月	<a href="#">管理</a>   <a href="#">远程连接</a> <a href="#">更改实例规格</a>   <a href="#">更多</a>

登录服务器查看 `/var/log/MatchvsTool.log` 日志，初始化大约在6-10分钟左右，初始化完成后，会进行安装测试，测试的结果会在日志最下方显示出来。如下显示：

```
[root@iZ2ze0qcppq4bbezc8dvtZ ~]# tailf /var/log/MatchvsTool.log
[2018-10-15 09:44:14,112]:[Init]:439:DEBUG:加载虚拟网卡
[2018-10-15 09:46:14,215]:[Init]:442:DEBUG:获取外网IP
[2018-10-15 09:46:14,234]:[Init]:444:DEBUG:InternetAddress 47.95.113.113
[2018-10-15 09:46:14,234]:[Init]:446:DEBUG:获取实例IP
[2018-10-15 09:46:14,241]:[GetInstanceID]:81:DEBUG:url:http://100.100.100.200/latest/meta-data/instance-id ret:i-2ze0qcppq4bbezc8dvt
[2018-10-15 09:46:14,242]:[Init]:448:DEBUG:instanceID: i-2ze0qcppq4bbezc8dvt
[2018-10-15 09:46:14,242]:[Init]:450:DEBUG:生成实例ID
[2018-10-15 09:46:14,628]:[Init]:452:DEBUG:license: 01006f203bi-2ze0qcppq4bbezc8dvt00000005bc3f16674cfe5ed500e4450f392f183fad**1v** #隐藏测试License
[2018-10-15 09:46:14,648]:[Init]:456:DEBUG:配置Hotel Sdk地址
[2018-10-15 09:46:14,648]:[ModifyHotelAddress]:352:INFO:修改hotel地址:
[2018-10-15 09:46:14,652]:[ModifyHotelAddress]:355:DEBUG:hotel-config-9330
[2018-10-15 09:46:14,656]:[ModifyHotelAddress]:360:DEBUG:47.95.113.113:9330
[2018-10-15 09:46:14,659]:[cleanup]:343:DEBUG:清理redis数据!!
```

```
[2018-10-15 09:46:14,847]:[ExecPod]:339:DEBUG:OK

[2018-10-15 09:46:14,848]:[cleanup]:349:INFO:OK

[2018-10-15 09:46:15,013]:[ExecPod]:339:DEBUG:OK

[2018-10-15 09:46:15,014]:[cleanup]:349:INFO:OK

[2018-10-15 09:46:15,170]:[ExecPod]:339:DEBUG:OK

[2018-10-15 09:46:15,170]:[cleanup]:349:INFO:OK

[2018-10-15 09:46:15,337]:[ExecPod]:339:DEBUG:READONLY You can't write against a read only
slave.

[2018-10-15 09:46:15,338]:[cleanup]:349:INFO:READONLY You can't write against a read only slave.

[2018-10-15 09:46:15,526]:[ExecPod]:339:DEBUG:READONLY You can't write against a read only
slave.

[2018-10-15 09:46:15,526]:[cleanup]:349:INFO:READONLY You can't write against a read only slave.

[2018-10-15 09:46:15,692]:[ExecPod]:339:DEBUG:READONLY You can't write against a read only
slave.

[2018-10-15 09:46:15,692]:[cleanup]:349:INFO:READONLY You can't write against a read only slave.

[2018-10-15 09:46:15,692]:[CleanZookeeperData]:366:DEBUG:清理zookeeper数据！！
[2018-10-15 09:46:15,854]:[ExecPod]:339:DEBUG:/bin/rm: cannot remove
'/var/lib/zookeeper/data/log/version-2': Directory not empty

[2018-10-15 09:46:56,057]:[ExecPod]:339:DEBUG:/bin/rm: cannot remove
'/var/lib/zookeeper/data/log/version-2': Directory not empty

[2018-10-15 09:47:36,229]:[ExecPod]:339:DEBUG:/bin/rm: cannot remove
'/var/lib/zookeeper/data/log/version-2': Directory not empty

[2018-10-15 09:48:16,275]:[EngineRestart]:375:INFO:重启引擎中！！
[2018-10-15 09:48:16,310]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-gosnowflake-
5fc57d8874-k42x1
[2018-10-15 09:48:18,328]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-gateway-77fbc989b9-
hq8gw
[2018-10-15 09:48:20,336]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-statistics-
5d4446c745-btfds
[2018-10-15 09:48:22,343]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-matchvsadmin-
6cb4f8ff5d-8jg9s
[2018-10-15 09:48:24,354]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-hotel-9330-
```

```

5454f4bd7-thgsh
[2018-10-15 09:48:26,362]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-directory-bb59d5fb-
k4c1h
[2018-10-15 09:48:28,374]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-dispatcher-
68d9cd6d9c-cvknb
[2018-10-15 09:48:30,384]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-beats-6d4bf94949-
wjgf5
[2018-10-15 09:48:32,393]:[EngineRestart]:386:INFO>Delete Engine POD: deploy-mvs-b7b4bc67f-68cjj
[2018-10-15 09:49:55,177]:[ShowEngine]:403:INFO:NAME READY
STATUS RESTARTS AGE
deploy-beats-6d4bf94949-b4cwt 3/3 Running 0 1m
deploy-directory-bb59d5fb-qt2t5 3/3 Running 2 1m
deploy-dispatcher-68d9cd6d9c-g7jfc 3/3 Running 0 1m
deploy-gateway-77fbc989b9-qkrhw 3/3 Running 0 1m
deploy-gosnowflake-5fc57d8874-g9jw2 3/3 Running 0 1m
deploy-hotel-9330-5454f4bd7-zkjrf 3/3 Running 0 1m
deploy-matchvsadmin-6cb4f8ff5d-qxrww 1/1 Running 0 1m
deploy-mvs-b7b4bc67f-7r5t8 3/3 Running 0 1m
deploy-statistics-5d4446c745-k4k5k 3/3 Running 0 1m

[2018-10-15 09:49:55,179]:[testEngine]:411:DEBUG:cd /data/zhangwan/frigate/bin && ./frigate -
gameid=2011226 -appkey=b90b138ab59f4e289fbd58182d4187bb -
appsecret=bf53a5a8b29c4198abe0e71c9e05d465 -gatewayaddr=172.17.187.12:7001 -
wss_proxy=wss://wss.local -nettype=0 -gs=0 -stop=1
[2018-10-15 09:50:12,392]:[testEngine]:415:ERROR:2018/10/15 09:49:55 proto: duplicate proto type
registered: stream.ProtoHeader

[2018-10-15 09:50:12,392]:[testEngine]:416:INFO:调用总次数:2
CmdID: joinOver 成功:2 失败:0
CmdID: broadCast 成功:2 失败:0
CmdID: subsCribе 成功:2 失败:0
CmdID: heartbeatToGateway 成功:2 失败:0
CmdID: joinRoom 成功:2 失败:0
CmdID: getRoomList 成功:1 失败:0
CmdID: waitkick 成功:2 失败:0
CmdID: checkIn 成功:2 失败:0
CmdID: publish 成功:2 失败:0
CmdID: logout 成功:2 失败:0
CmdID: login 成功:2 失败:0
CmdID: leaveRoom 成功:2 失败:0
CmdID: createRoom 成功:1 失败:0
CmdID: kickPlayer 成功:1 失败:0

```

当安装测试完成日志输出后，则表示 Matchvs 引擎已经可以正常使用。Matchvs引擎是基于 `kubernetes 1.8.5` 版本上的程序，所以kubernetes 所具有的功能都能正常使用。对于以下域名，可根据需要添加本地hosts来访问对应的服务，`websocket` 域名可以通过 `MatchvsTool` 工具来进行增加：

(可选) kubernetes dashboard 访问域名：`dashboard.local`

(可选) kubernetes grafana 访问域名：`grafana.local`

## 添加证书 (可选)

如果游戏需要使用 wss 协议（如微信小游戏），则需要完成此步骤。

Matchvs 提供了命令行工具（以下简称引擎工具）进行协议及游戏信息的添加。注意：MatchvsTool 依赖于 `/root/.kube/config` 文件来进行访问 kubernetes，请勿删除

```
usage: MatchvsTool [-h] [-s] [-l] [-c license] [-t] [-w domain cert key]
                  [-a gameID gameName appKey appSecret] [-d gameID]
                  [-m gameID gameName] [-g] [--restart] [--instance-id]
```

Matchvs 引擎工具

optional arguments:

- `-h, --help` show this help message and `exit`
- `-s` 查看当前引擎状态,以及域名信息
- `-l` 查看当前引擎使用的license
- `-c license` 修改引擎license
- `-t` 测试引擎
- `-w domain cert key` 添加wss域名与证书到引擎中
- `-a gameID gameName appKey appSecret` 增加游戏到引擎中
- `-d gameID` 删除指定gameID的游戏
- `-m gameID gameName` 修改指定gameID的游戏名称
- `-g` 列出当前引擎有那些gameID等信息
- `--restart` 重启引擎
- `--instance-id` 显示当前阿里云主机实例ID

SSL证书的获取方法可参考网上教程：<https://blog.csdn.net/luckymama/article/details/64904729>

使用 `MatchvsTool -w domain cert key` 参数来把指定的域名添加到Matchvs 里面。

**注意证书路径请写全路径。**

操作如下显示：

```
[root@iZ2ze0wn45z3cf6adel14zZ ~]# MatchvsTool -w aligateway.matchvs.com /root/matchvs.com.crt /root/matchvs.com.key
```

添加SSL证书成功!!!，引擎将会重启！！

```
Delete Engine POD: deploy-gosnowflake-5fc57d8874-v2g4m
Delete Engine POD: deploy-gateway-77fbc989b9-qt4kn
Delete Engine POD: deploy-statistics-5d4446c745-jm27t
Delete Engine POD: deploy-matchvsadmin-6cb4f8ff5d-h98z6
Delete Engine POD: deploy-hotel-9330-5454f4bd7-bqnvj
Delete Engine POD: deploy-directory-bb59d5fb-ssmsm
Delete Engine POD: deploy-dispatcher-68d9cd6d9c-pngzc
Delete Engine POD: deploy-beats-6d4bf94949-m8xkg
Delete Engine POD: deploy-mvs-b7b4bc67f-5xtnl
```

NAME	READY	STATUS	RESTARTS	AGE
deploy-beats-6d4bf94949-hrj27	3/3	Running	0	2m
deploy-directory-bb59d5fb-gtsz8	3/3	Running	0	2m
deploy-dispatcher-68d9cd6d9c-wckjl	3/3	Running	0	2m
deploy-gateway-77fbc989b9-psnpp	3/3	Running	0	2m
deploy-gosnowflake-5fc57d8874-tsd6d	3/3	Running	0	2m
deploy-hotel-9330-5454f4bd7-xphlt	3/3	Running	0	2m

deploy-matchvsadmin-6cb4f8ff5d-6whvm	1/1	Running	0	2m
deploy-mvs-b7b4bc67f-1m65c	3/3	Running	0	2m
deploy-statistics-5d4446c745-8zj59	3/3	Running	0	2m

## 添加游戏

在使用 Matchvs 客户端 SDK 及 gameServer 框架开发联网功能时，需要用到此处添加的游戏信息。

gameID 是游戏ID，gameName 是游戏名称，appkey 和 appSecret 是游戏密钥信息，均根据开发需求进行自定义。

使用 `MatchvsTool -a $gameID $游戏名称 $appkey01 $appsecret01` 将游戏添加到 Matchvs里。请注意替换"\$变量"

添加成功后，会输出当前引擎里面有多少个游戏。

**注意：frigate 是用于测试 Matchvs 的游戏信息，不可以删除。**

```
MatchvsTool -a 1 lvttest appkey01 appsecret01
gameID:1   gameName:lvttest appKey:appkey01 appSecret:appsecret01
gameID:201226 gameName:frigate appKey:b90b138ab59f4e289fbd58182d4187bb
appSecret:bf53a5a8b29c4198abe0e71c9e05d465
```

至此，您已完成全部安装步骤啦！您可以前往 Matchvs 官网服务中心查看联网功能如何接入。

## web 控制台

Matchvs 提供了 web 控制台可以查看游戏数据监控信息、许可证信息。

开通服务后，进入阿里云买家控制台，可以直接进入 Matchvs web 控制台：







## Matchvs 工具其他重要功能

### 查看游戏信息

-g 参数，可以列出所有在引擎添加过的游戏信息。

```
[root@iZ2ze0wn45z3cf6ade114zZ ~]# MatchvsTool -g
gameID:1000 gameName:alitest appKey:ms111 appSecret:abc123
gameID:5000 gameName:lvtest appKey:b90b138ab5 appSecret:a8b29c4198
gameID:201226 gameName:frigate appKey:b90b138ab59f4e289fbd58182d4187bb
appSecret:bf53a5a8b29c4198abe0e71c9e05d465
```

### 删除游戏

-d gameId 可以删除指定gameID的游戏。首先使用-g 参数来显示当前引擎添加了多少游戏，再使用-d gameId 来删除指定游戏。

```
[root@iZ2ze74bqinfs514yb10pgZ ~]# MatchvsTool -g
gameID:201226  gameName:frigate      appKey:b90b138ab59f4e289fbd58182d4187bb
appSecret:bf53a5a8b29c4198abe0e71c9e05d465
gameID:202254  gameName:史莱姆 appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
gameID:202255  gameName:史莱姆2  appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
gameID:202256  gameName:史莱姆3  appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
[root@iZ2ze74bqinfs514yb10pgZ ~]# MatchvsTool -d 202256
gameID:201226  gameName:frigate      appKey:b90b138ab59f4e289fbd58182d4187bb
appSecret:bf53a5a8b29c4198abe0e71c9e05d465
gameID:202254  gameName:史莱姆 appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
gameID:202255  gameName:史莱姆2  appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
```

## 修改游戏名称

-m gameId gameName 可以方便的修改游戏名称

```
[root@iZ2ze74bqinfs514yb10pgZ ~]# MatchvsTool -g
gameID:201226  gameName:frigate      appKey:b90b138ab59f4e289fbd58182d4187bb
appSecret:bf53a5a8b29c4198abe0e71c9e05d465
gameID:202254  gameName:史莱姆 appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
gameID:202255  gameName:史莱姆2  appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
[root@iZ2ze74bqinfs514yb10pgZ ~]# MatchvsTool -m 202255 史姆菜5
gameID:201226  gameName:frigate      appKey:b90b138ab59f4e289fbd58182d4187bb
appSecret:bf53a5a8b29c4198abe0e71c9e05d465
gameID:202254  gameName:史莱姆 appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
gameID:202255  gameName:史姆菜5  appKey:5147cceadc7446e48a38a411d2e8f4a0#M
appSecret:c85e7d6acc1241b992c1c836c69635ff
```

## 查看运行状态

-s 参数会显示引擎 的状态以及域名列表，数据库信息等

```
[root@iZ2ze0wn45z3cf6adel114zZ ~]# MatchvsTool -s
加载部署信息：

域名列表：
dashboard.local:  *****
test1111.matchvs.com:  *****
grafana.local:  *****

数据库地址：
host:  *****
```

```
port: *****
pwd: *****

version: v3.8.1.3
```

NAME	READY	STATUS	RESTARTS	AGE
deploy-beats-6d4bf94949-hrj27	3/3	Running	0	12m
deploy-directory-bb59d5fb-gtsz8	3/3	Running	0	12m
deploy-dispatcher-68d9cd6d9c-wckj1	3/3	Running	0	12m
deploy-gateway-77fbc989b9-psnpp	3/3	Running	0	12m
deploy-gosnowflake-5fc57d8874-tsd6d	3/3	Running	0	12m
deploy-hotel-9330-5454f4bd7-xphlt	3/3	Running	0	12m
deploy-matchvsadmin-6cb4f8ff5d-6whvm	1/1	Running	0	12m
deploy-mvs-b7b4bc67f-1m65c	3/3	Running	0	12m
deploy-statistics-5d4446c745-8zj59	3/3	Running	0	12m

## 查看授权信息

license 是 Matchvs 提供的服务授权使用证书，每个 license 都有可允许使用的用量。

CCU 是指同时在线人数，据行业经验估算：日活  $\approx$  CCU \* 20

在阿里云市场直接购买的是 20 CCU 的证书：

```
[root@iZ2ze0wn45z3cf6adel14zZ ~]# MatchvsTool -l
license:01006f203bi-2ze0wn45z3cf6adel14z0000000***** ccu:20 type:0
```

## 修改授权信息

-c 参数允许用户修改license。更换 license 请到 Matchvs 官网去申请，然后使用 MatchvsTool 工具来进行修改。修改 license 后可以使用 -l 参数来进行验证。

```
[root@iZ2ze0wn45z3cf6adel14zZ ~]# MatchvsTool -c 100000296600000005b671*****
修改引擎license成功！！
```

## 测试安装结果

-t 参数，会调用 /data/zhangwan/frigate/bin/frigate 程序，来进行对 Matchvs 的简单测试，方便用户测试引擎的可用性。

```
MatchvsTool -t
2018/10/17 15:12:51 proto: duplicate proto type registered: stream.ProtoHeader

调用总次数:2
CmdID:          login 成功:2    失败:0
CmdID:          leaveRoom 成功:2    失败:0
CmdID:          broadCast 成功:2    失败:0
CmdID:          subsCribе 成功:2    失败:0
CmdID:          waitkick 成功:2    失败:0
CmdID:          checkIn 成功:2    失败:0
```

CmdID:	publish	成功:2	失败:0
CmdID:	logout	成功:2	失败:0
CmdID:	heartbeatToGateway	成功:2	失败:0
CmdID:	joinRoom	成功:2	失败:0
CmdID:	joinOver	成功:2	失败:0
CmdID:	getRoomList	成功:1	失败:0
CmdID:	createRoom	成功:1	失败:0
CmdID:	kickPlayer	成功:1	失败:0

## 其他参数说明

<code>--restart</code>	重启引擎
<code>--instance-id</code>	显示当前阿里云主机实例ID

`restart` 参数，用于手动重启引擎。当引擎异常时，可以尝试执行引擎重启来恢复引擎故障。

`instance-id` 参数，是阿里云当前 ECS 实例 ID，方便用户后期替换 license 时，提供给 Matchvs 用于确定机器。

通过学习开通云市场镜像版的教程，你应该已经成功安装服务并添加游戏信息了。

接下来，我们通过运行Demo，来实际了解在镜像里添加的相关信息的作用。

## Demo 体验

在三个不同的浏览器(如，chrome,Opera,IE)打开 [Demo 在线体验地址](http://demo.matchvs.com/Egret/) : <http://demo.matchvs.com/Egret/>，点击页面右下角“独立部署”，进入独立部署 Demo。

The screenshot shows the MatchvsDemo-Egret interface. It features three input fields for 'gameID', 'appKey', and 'secretKey'. Below these fields are two buttons: '进入Demo' (Enter Demo) and '清除缓存' (Clear Cache). At the bottom, there is a提示 (提示) section and a red-bordered button labeled '独立部署Demo' (Independent Deployment Demo).

MatchvsDemo-Egret

gameID 201489

appKey 4fb6406305f44f1aad0c40e5946ffe3d

secretKey 5035d62b75bd4941b182579f2b8fc12c

进入Demo 清除缓存

提示: 此Demo需要分别在三个不同的浏览器 (如Chrome、火狐、360) 里打开进行体验  
如需在一个浏览器里完成体验, 则每次需要先“清除缓存”

独立部署Demo

## Matchvs独立部署

服务地址:	<input type="text" value="mt21gateway.matchvs.com"/>
gameID:	<input type="text" value="1"/>
appKey:	<input type="text" value="appkey01"/>
secretKey:	<input type="text" value="appsecret01"/>
userID:	<input type="text" value="123456"/>
token:	<input type="text" value="OEWIURIOJNUOGIUDSF809LJOKETGT89H"/>

Matchvs提供联网与服务端能力

- 以上信息就是我们在云市场镜像里，通过 MatchvsTool 添加的域名、游戏信息。
- 服务地址，为域名信息，在云市场里镜像教程"添加证书"部分有介绍。该域名，为 `MatchvsTool -w domain cert key` 里的 `demain` 信息。
- `gameID, appKey, secretKey` ，为游戏信息，在云市场镜像教程"添加游戏"部分有介绍。如，在镜像里执行 `MatchvsTool -a 1 lvtest appkey01 appsecret01` ，则上述输入框，分别填写gameID: `1` ;appKey : `appkey01` ，secretKey : `appsecret01`
- userID 可以自定义填写，**注意：三个浏览器里的 userID 不要重复**
- token 为用户令牌，可自定义填写。
- 分别点击登录，即可进入大厅：



- 分别点击随机匹配，即可三个玩家匹配到一起开始游戏。

# 接入说明

Matchvs 提供了三种外放形态：云市场镜像、独立部署、Matchvs 游戏云托管。

**云市场镜像**是指，直接通过阿里云开通服务，购买服务器后，整套 Matchvs 服务自动部署完成。

**独立部署**是指，开发者用自己的服务器，或购买云服务器，然后手动安装整套 Matchvs 服务。

**Matchvs 游戏云托管**，是指无需购买任何机器，整套服务托管在 Matchvs ，开发者直接接入联网 SDK 即可。

**镜像版与独立部署在联网功能接入时的流程及方法完全相同。**

**镜像版（独立部署）与使用 Matchvs云托管在接联网 SDK不同之处是：初始化接口不同，以及不需要调用 `registerUser` 接口，`userID` 和 `token`是由开发者自己定义。**

镜像版使用的初始化接口是 `premiseInit` 。初始化回调接口与云托管相同，使用 `initResponse` 接口。

登录以及其他操作都是使用与 MatchvsSDK 云游戏相同，教程请参考：[creator 教程](#)

:<http://www.matchvs.com/service?page=CocosCreatorStart> ,

[Egret 教程](#) :<http://www.matchvs.com/service?page=EgretStart>。

如需了解更多 Demo 实现细节，请参考此[Demo 教程](https://github.com/matchvs/demo-Egret) :<https://github.com/matchvs/demo-Egret>

初始化说明：MsEngine.ts 和 MsResponse.ts 文件定义

```
//MsEngine.ts 请求
public premiseInit(endPoint:string, gameId:number):number{
    this._response = MsResponse.getInstance.getResponse();
    let res = this._engine.premiseInit( MsResponse.getInstance.getResponse(),endPoint, gameId);
    if (res !== 0){
        console.info("[MsEngine premiseInit failed] resCode:",res);
        return res;
    }
    console.info("[MsEngine premiseInit success] resCode:",res);
    return res;
}

//MsResponse.ts 回调
private initResponse(status:number){
    console.info("initResponse status:",status);
    this.dispatchEvent(new egret.Event(MsEvent.EVENT_INIT_RSP,false,false,{status:status}));
}
```

使用示例

```
//PremiseLoginUI.ts
```

```

button.addEventListener(egret.TouchEvent.TOUCH_TAP, e => {
    GameData.configEnvir(input.text, cbx.selected);
    console.log(" environment=" + GameData.DEFAULT_ENV + " gameid=" + GameData.gameID);
    //这里调用 MsEngine.ts 中的函数
    let result = mvs.MsEngine.getInstance.init(GameData.CHANNEL, GameData.DEFAULT_ENV,
    GameData.gameID);
        }, this);
/**
 * 初始化
 */
private premiseInit(event:egret.TouchEvent){
    let endPoint:string = this.txt_endport.text;
    let gameID:number = Number(this.txt_gameID.text);
    mvs.MsEngine.getInstance.premiseInit(endPoint,gameID);
}

//LoginView.ts 在页面用户监听到的回调事件
private initResponse(ev:egret.Event) {
    console.log("initResponse,status:" + ev.data.status);
    ...
}

```

**注意 在整个应用全局，开发者只需要对 Matchvs 做一次初始化。**