
ESSS 服务软件 V2.0.1 用户操作手册

北京西部世界科技有限公司

编制单位：北京西部世界科技有限公司

日期：2023 年 8 月 15 日

目录

1. 软件安装及登录	1
1.1. 下载安装	1
1.2. 账号登录	1
1.3. 修改账号密码	2
2. 系统首页	3
3. 系统授权	4
3.1. 机器码获取	4
3.2. 授权申请	5
3.3. 更新授权	5
3.3.1. 自动更新	6
3.3.2. 手动更新	6
4. 应用配置	7
4.1. 增加应用	7
4.2. 配置参数	8
4.3. 修改应用	9
4.4. 启动应用	10
5. 正在运行	11
6. 静态场景	13
6.1. 添加静态场景	13
6.2. 编辑静态场景	13
6.2.1. 新建文件夹	14
6.2.2. 新建 ESUnrealActor 对象	14
6.2.3. 恢复初始视角	15
6.2.4. 导入文件	15
6.2.5. 视角管理	16
6.3. 修改静态场景 json	16
7. 资源监控	18
8. 系统配置	19
8.1. 启动端口配置	19
8.2. 实例个数配置	19
8.3. 自动关闭实例	20
8.4. GPU 剩余量限制	21
9. 系统日志	22
10. UE 云渲染通信流程	23

1.1 引入 esueviewer.js 文件	23
1.2 通过 websocket 服务建立通信关系	24
10.1.1. 拼接获取连接 UE 实例的请求地址	24
10.1.2. 创建 UEViewer	25
1.3 js 监听接收 UE 消息	25
1.4 js 发送消息通信 UE	26

1. 软件安装及登录

1.1. 下载安装

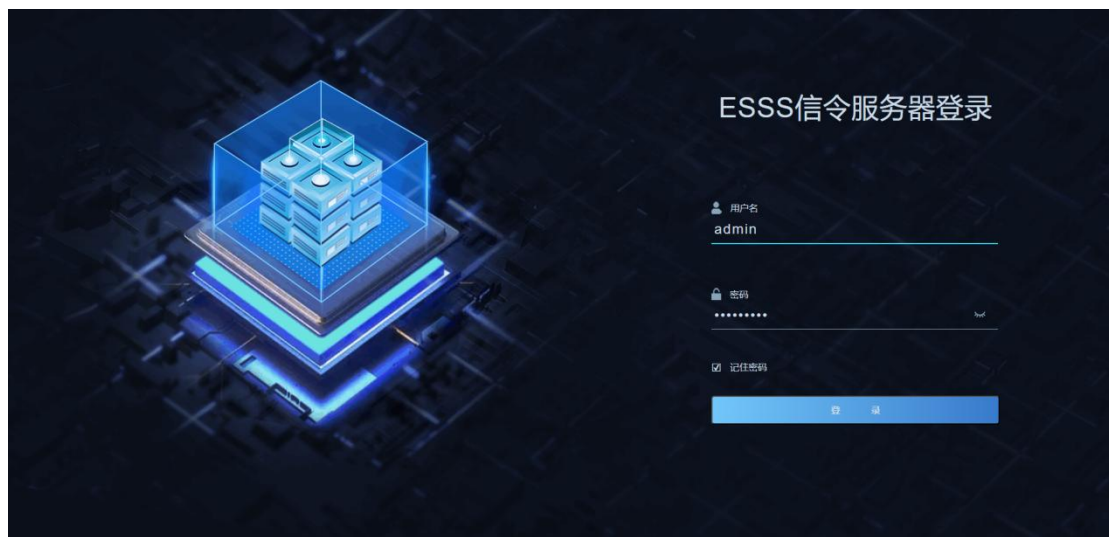
<http://www.cesiumlab.com/esss.html> 官网下载安装软件

1.2. 账号登录

<http://localhost:8086> 跳转进入登录页面

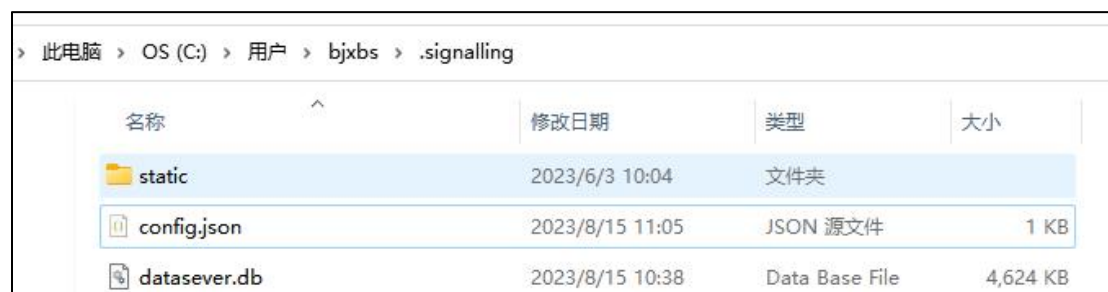
具体修改账号密码看【[1.3 修改账号密码](#)】

管理员	
账号	admin
密码	admin1234



1.3. 修改账号密码

【ESSS 信令服务】只有一个用户即【管理员用户】，该管理员的账号密码以及其他配置信息存放到 C 盘用户目录下.signalling 文件夹下的 config.json



```
{
  "minsocketport":8087,
  "port":8086,
  "maxinstance":1,
  "turnserverport":8085,
  "host": "*",
  "admin":{
    "password": "admin1234",
    "username": "admin"
  },
  "isAutoOffInstance":true,
  "maxsocketport":9999
}
```

字段名	字段含义
minsocketport	启动云渲染连接的最小端口号
port	ESS 信令服务的客户端端口号
maxinstance	最大实例个数
trunserverport	中继转发服务端端口号
password	管理员密码
username	管理员用户名
isAutoOffInstance	无用户连接实例是否自动关闭
maxsocketport	启动云渲染连接的最大端口号

2. 系统首页

【ESSS 信令服务】是【UE 云渲染程序】管理，系统首页包含整个服务的整体信息内容：最大实例个数、正在运行实例个数、应用总个数、CPU 资源占用、GPU 资源占用、传输速度、授权信息



3. 系统授权

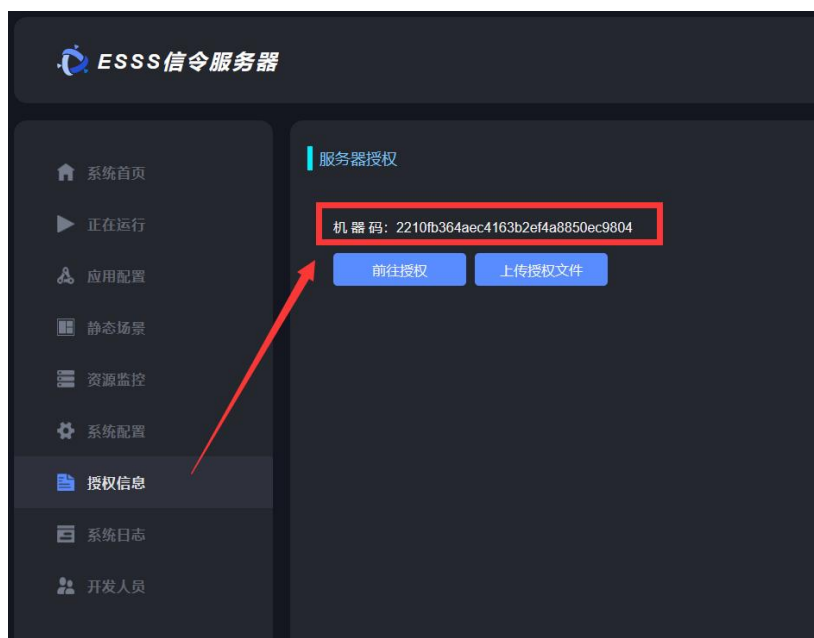
系统授权决定【实例个数】限制，无授权用户实例个数为 1，可测试以及使用 ESSS 信令服务的功能，但如果需求实例>1 则需联系官方客服购买授权，同时申请授权文件。

权限	云渲染实例个数
无授权	1 个实例
有授权	3、6、9、12 个实例

3.1. 机器码获取

【授权信息】中可以查看【ESSS 信令服务】的产品【机器码】

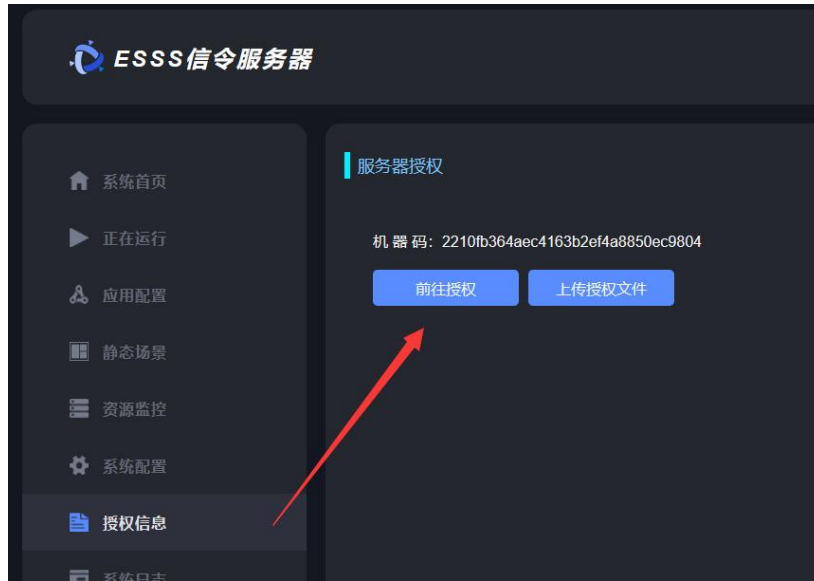
【机器码作用】根据机器码在西部世界云服务，申请对应的产品授权信息



3.2. 授权申请

温馨提示：以下授权申请操作是已经联系官方客服购买支付成功，需要申请授权文件

【授权信息】中点击【前往授权】，自动跳转到[地球可视化实验室用户后台](#)，根据流程进行授权申请即可



3.3. 更新授权

通过[地球可视化实验室用户后台](#)，可获取对应的授权文件(.lic 格式)，更新到指定目录下

【授权用户】分为两大类

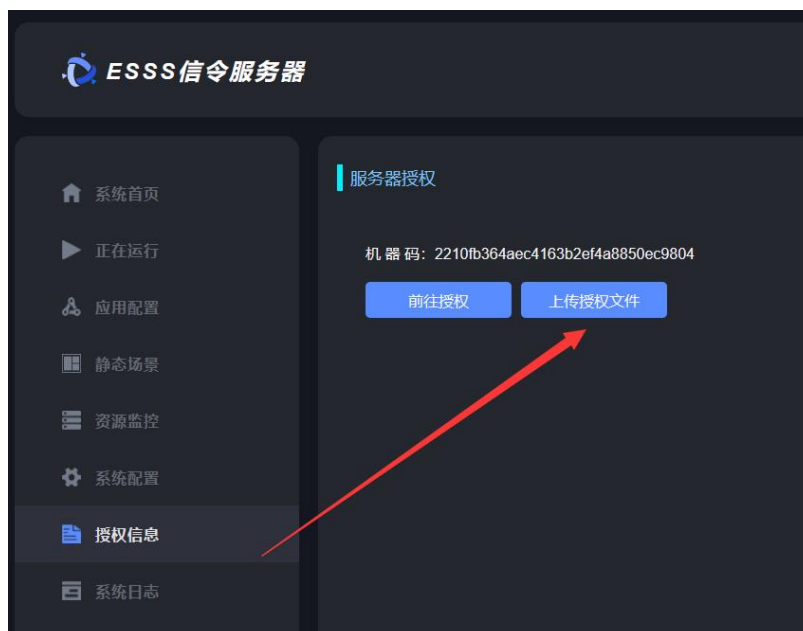
授权	区分
年版授权	1 年
永久授权	没有时间限制

【授权更新】分为以下两种方式：

授权更新方式	区分
自动更新	本机部署
手动更新	本机部署/服务器部署

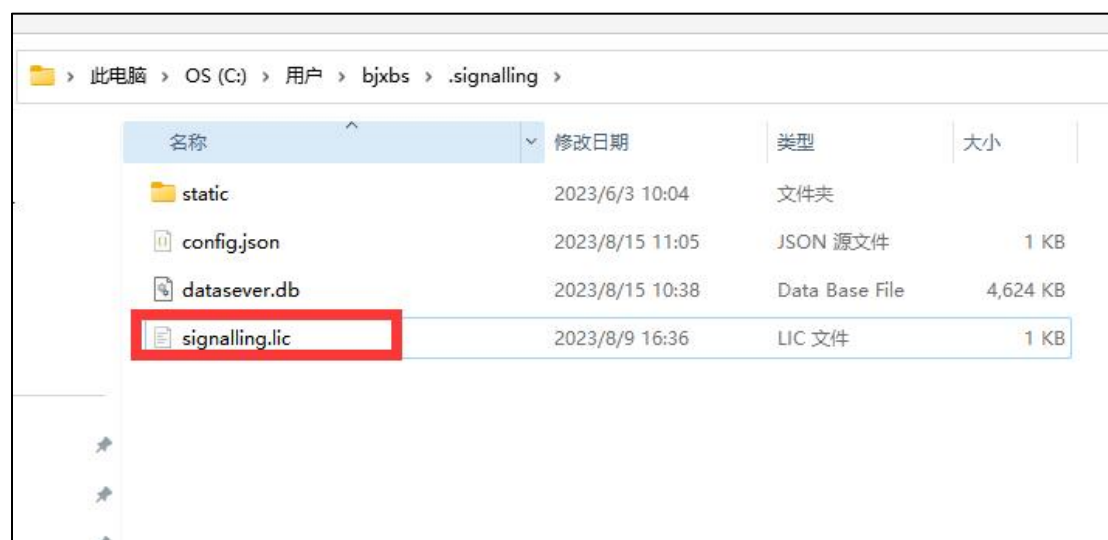
3.3.1. 自动更新

【授权信息】如果访问的是本地服务器可以通过点击【上传授权文件】弹出文件选择器，选择获取的授权文件位置点击打开即可



3.3.2. 手动更新

【拷贝】授权文件到—>C 盘用户目录下 signalling 文件夹下，授权文件名称需要改成 signalling.lic



4. 应用配置

4.1. 增加应用

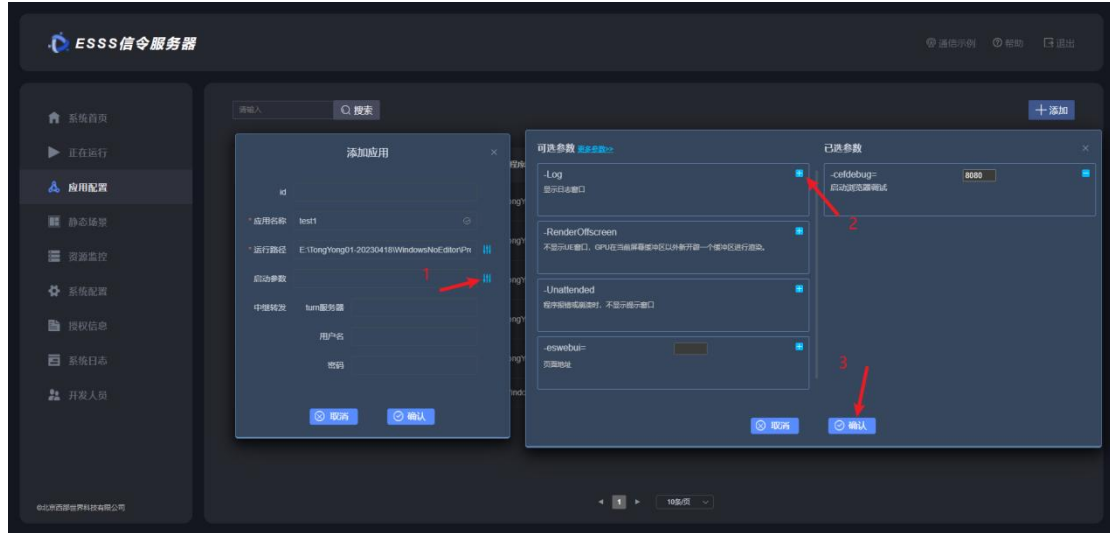
点击右上角【添加按钮】，弹出【添加应用】窗口如下图所示



字段	字段含义	是否必填项
Id	应用的 id	
应用名称	应用的名称	必填
运行路径	应用的运行路径	必填
启动参数	启动实例需要的参数	
turn 服务器	外网穿透	
用户名	中继转发的用户名	
密码	中继转发的密码	

4.2. 配置参数

UnrealEngine 程序有各种命令行参数，这些命令行参数不但可以程序内部利用 Execute Console Command 蓝图节点调用，很多参数可以启动时候携带命令行启动，因此实现了不同的功能，pixel 像素流云渲染启动也避免不了各种各样的参数，比如-log 显示日志、-Unattended 程序报错或崩溃时，不显示提示窗口等等，因此该参数配置，就是配置该 UE 程序的启动参数

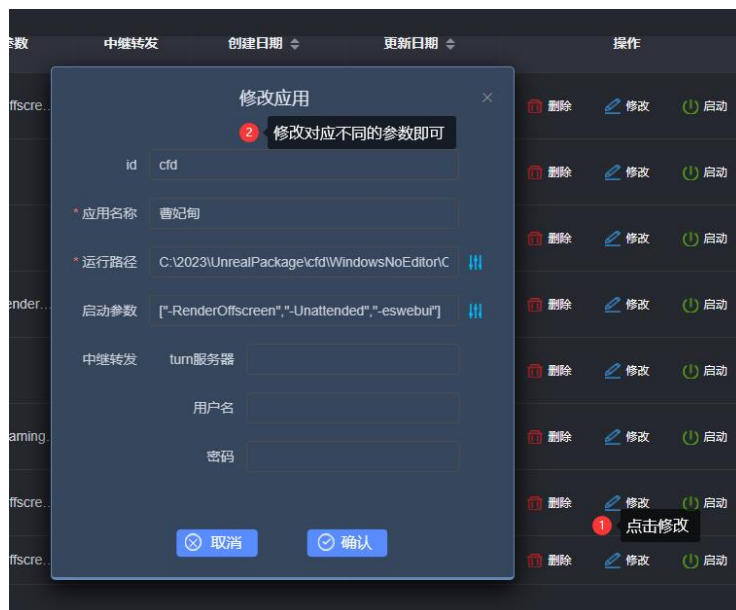


参数	参数含义
-Log	显示日志窗口
-RenderOffscreen	不显示 UE 窗口
-Unattended	程序报错或崩溃时，不显示提示窗口
-cefdebug=8085	使用插件 ESWebView 网页可调试端口
-eswebui=	使用插件 ESForUe 可加载该网页地址
-PixelStreamingHideCursor=	是否隐藏 UE 应用程序光标
-PixelStreamingEncoderTargetBitrate=	忽略 WebRTC 想要的比特率（不推荐）
-PixelStreamingEncoderMaxBitrate=	最大比特率（bps）
-PixelStreamingDebugDumpFrame=	将帧从编码器转储到磁盘上的文件以进行调试。
-PixelStreamingEncoderMinQP=	0-51 较低的值导致更好的质量，但更高的比特率。 注意-1 将禁用对最小 QP 的任何硬限制。
-PixelStreamingEncoderMaxQP=	0-51 较低的值导致更好的质量，但更高的比特率。 注意-1 将禁用对最小 QP 的任何硬限制。
-PixelStreamingEncoderRateControl=	像素流视频编码器速率控制模式
	通过填充垃圾数据来保持恒定的比特率。注意 CBR

-PixelStreamingEnableFillerData=	和 MinQP = -1 时不需要。
-PixelStreamingEncoderMultipass=	编码器每帧执行的传递次数。有关更多详细信息，
-PixelStreamingH264Profile=	编码器使用的 H264 配置文件。
-LogCmds=PixelStreamingWebRTC	指定 WebRTC 的日志级别对于调试 WebRTC 有用。
-PixelStreamingWebRTCDegradationPreference=	[INCLUDE:#DegredationPreferenceInfo]
-PixelStreamingWebRTCMaxFps=	WebRTC 将尝试捕获/编码/传输的最大 FPS
-PixelStreamingWebRTCStartBitrate=	WebRTC 将尝试开始流的起始比特率 (bps)。
-PixelStreamingWebRTCMinBitrate=	WebRTC 不会在下面请求的最低比特率 (bps)
-PixelStreamingWebRTCMaxBitrate=	WebRTC 不会在上面请求的最大比特率 (bps)。
-PixelStreamingWebRTCLowQpThreshold=	仅在 PixelStreaming.Encoder.LowQP=-1 时使用。介于 1 和 51 之间的值。如果 WebRTC 的帧低于这个 QP，它将尝试做出适应决定。
-PixelStreamingWebRTCHighQpThreshold=	仅在 PixelStreaming.Encoder.LowQP=-1 时使用
-PixelStreamingWebRTCDisableReceiveAudio=	禁止将音频从浏览器接收到 UE。
-PixelStreamingWebRTCDisableTransmitAudio=	禁用 UE 音频到浏览器的传输。
-PixelStreamingWebRTCDisableAudioSync=	禁用 WebRTC 中音频和视频轨道的同步。

4.3. 修改应用

点击列表中对应用的操作按钮【修改按钮】，弹出对应实例的配置信息，可进行手动修改，点击确认即可修改成功

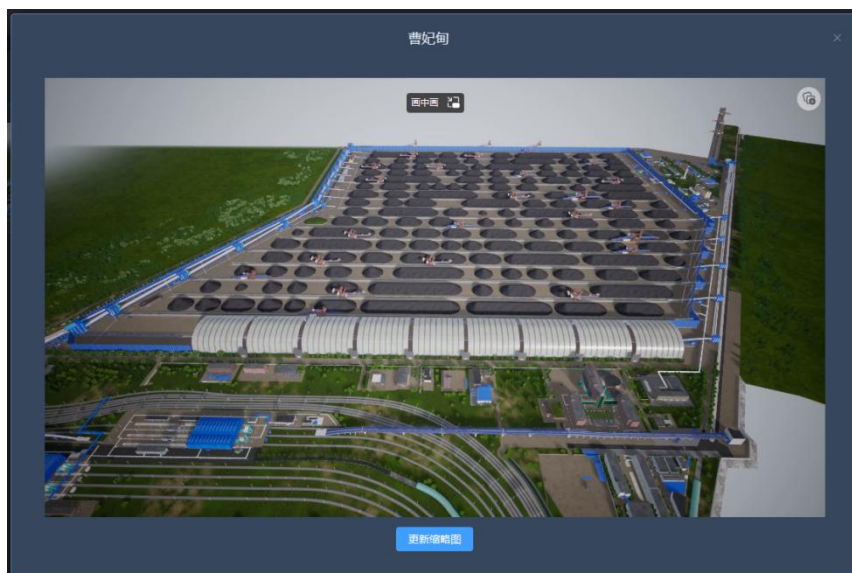


4.4. 启动应用

点击【启动】，启动成功页面会自动跳转到【正在运行】页面。

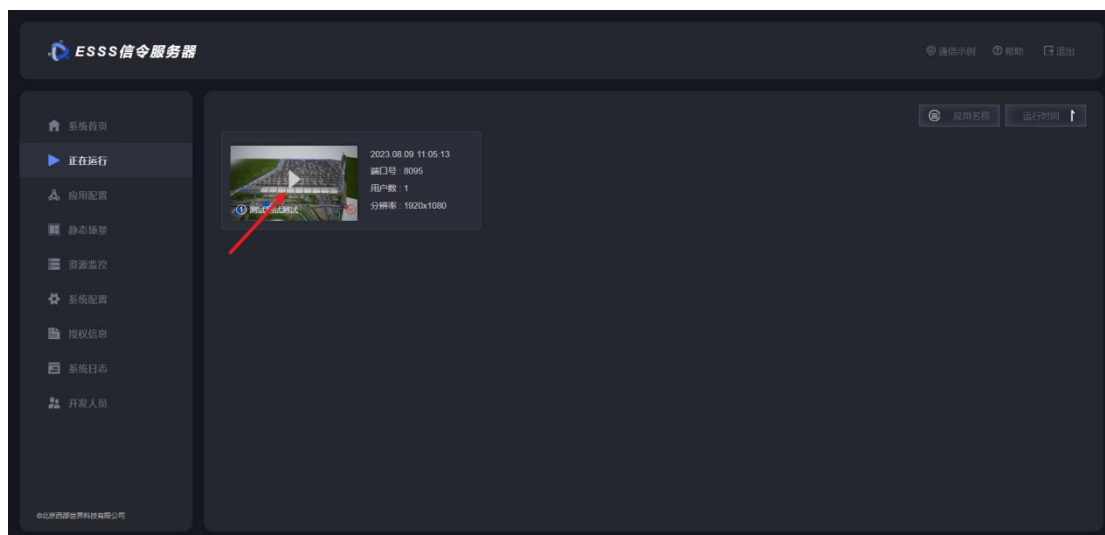


点击实例连接像素流视频画面，如下图所示

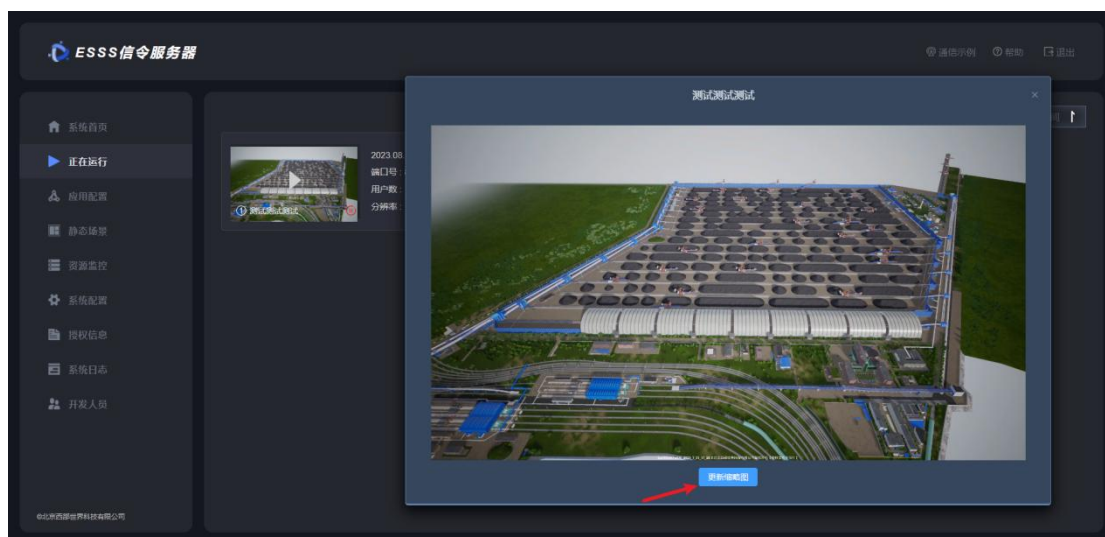


5. 正在运行

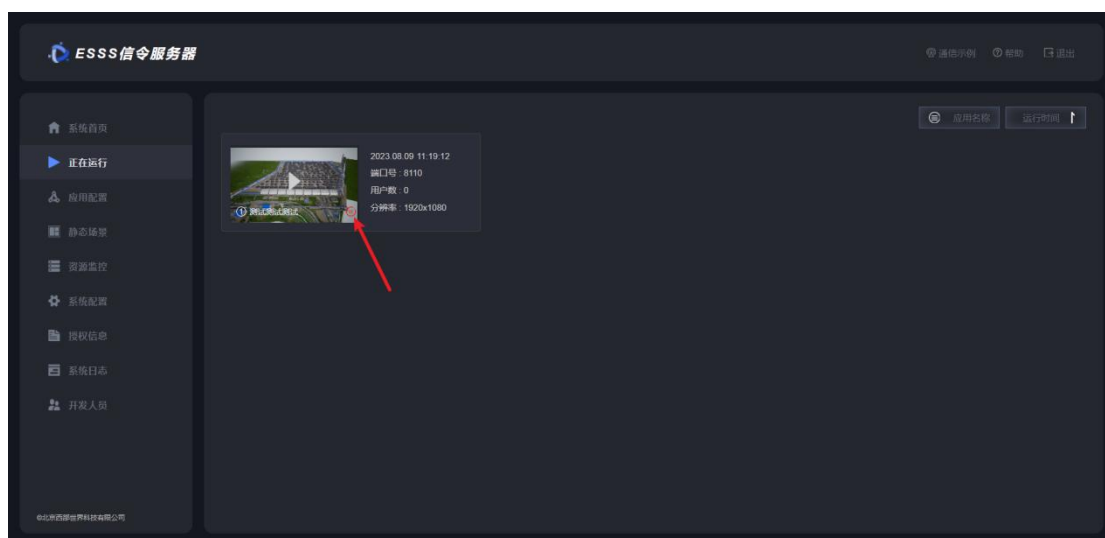
【正在运行】包括当前用户已启动的实例，显示缩略图，当前日期，端口号 and 用户数量。点击打开弹出框。



该弹出框提供播放功能和更新实时场景功能(注：此功能和 ESForUE 插件相关，并非支持所有)。



点击关闭实例，系统默认已开启自动关闭实例功能。在[系统配置](#)界面可修改。

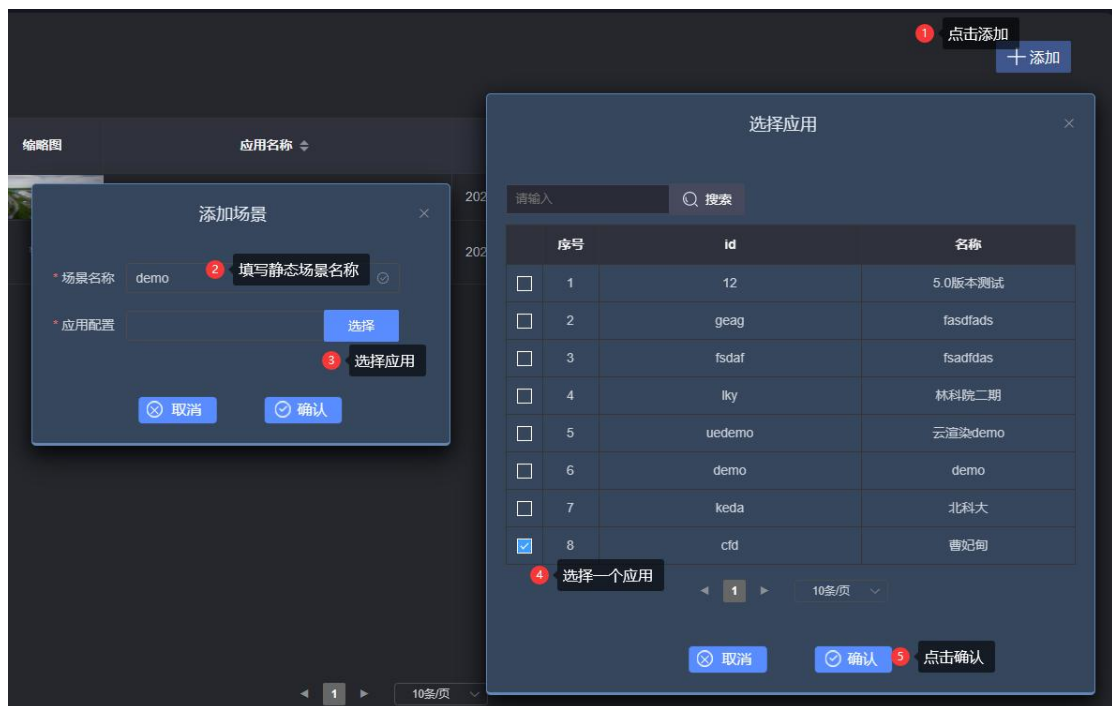


6. 静态场景

静态场景指的是, 对于一个场景的 Actor 进行 tag 的编辑标记最终生成一个 json 文件, 该 json 文件是 Earthsdk 的标准 json 格式, 用来开发使用 (注该功能需要 UE 引入 ESForUe 插件方可使用, 该插件会在后续放出, 请持续关注)

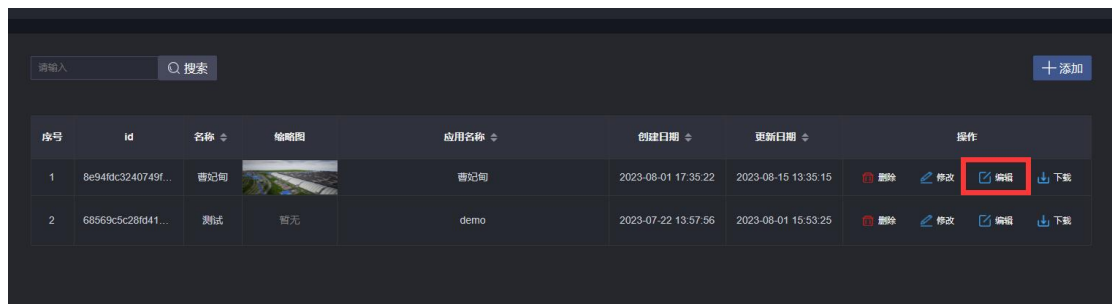
ESForUE 插件暂时没有推出, 请您持续关注, 该功能暂时略过

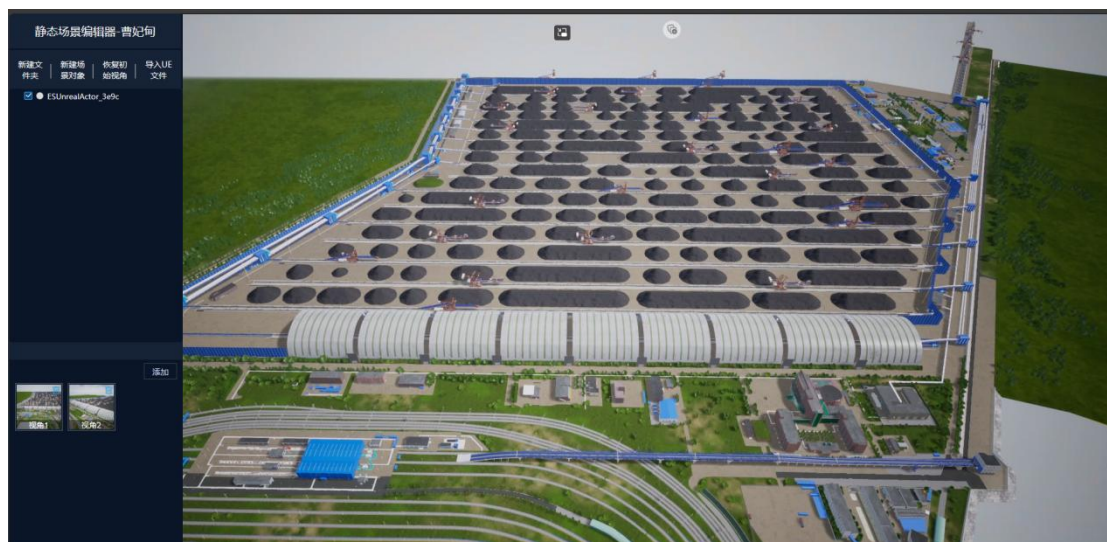
6.1. 添加静态场景



6.2. 编辑静态场景

点击【编辑】可跳转 tag 编辑器窗口, 如下图所示



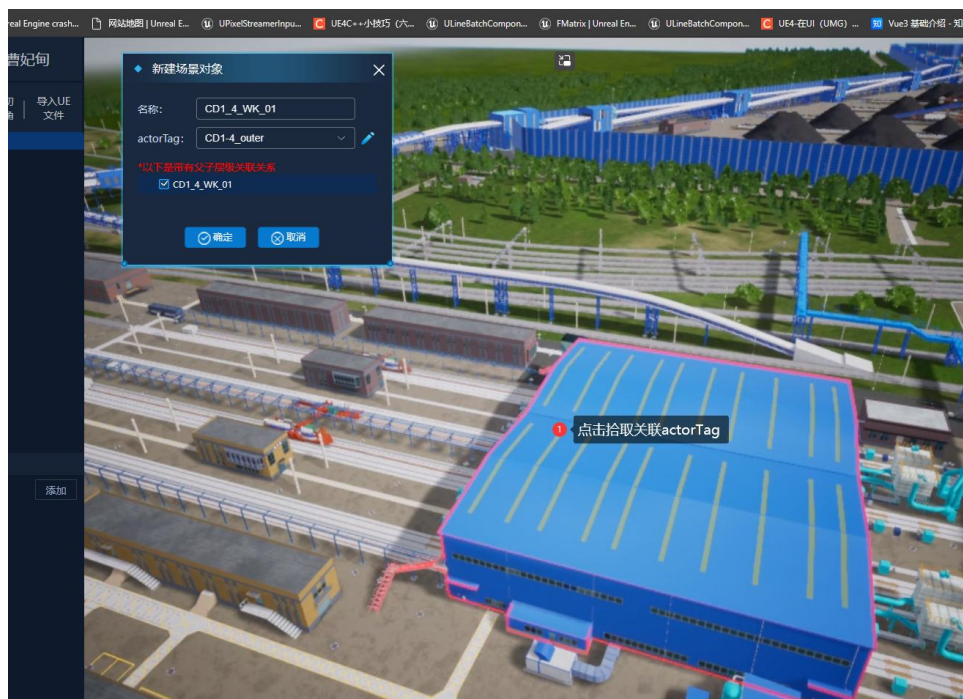


6.2.1. 新建文件夹



6.2.2. 新建 ESUnrealActor 对象

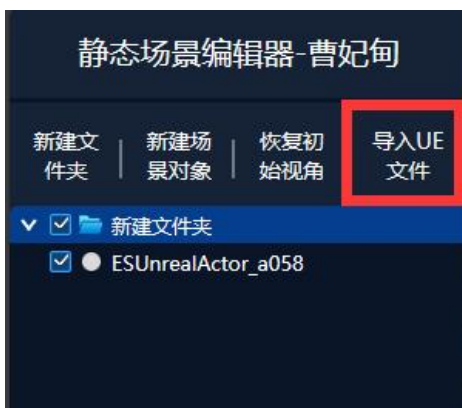




6.2.3. 恢复初始视角



6.2.4. 导入文件



6.2.5. 视角管理



6.3. 修改静态场景 json

由于静态场景编辑器就是为了导出一个标准的 Earthsdk 的 json 格式来加载，因此存储的就是一个 json 信息，所以该配置可以上传标准的 earthsdk 的 json 格式文件，以及手动编辑输入

修改方式	区分
上传文件	简单粗暴
手动编辑	可直接在 ESS 信令服务界面进行微调

点击【修改按钮】弹出修改场景窗口如下图所示

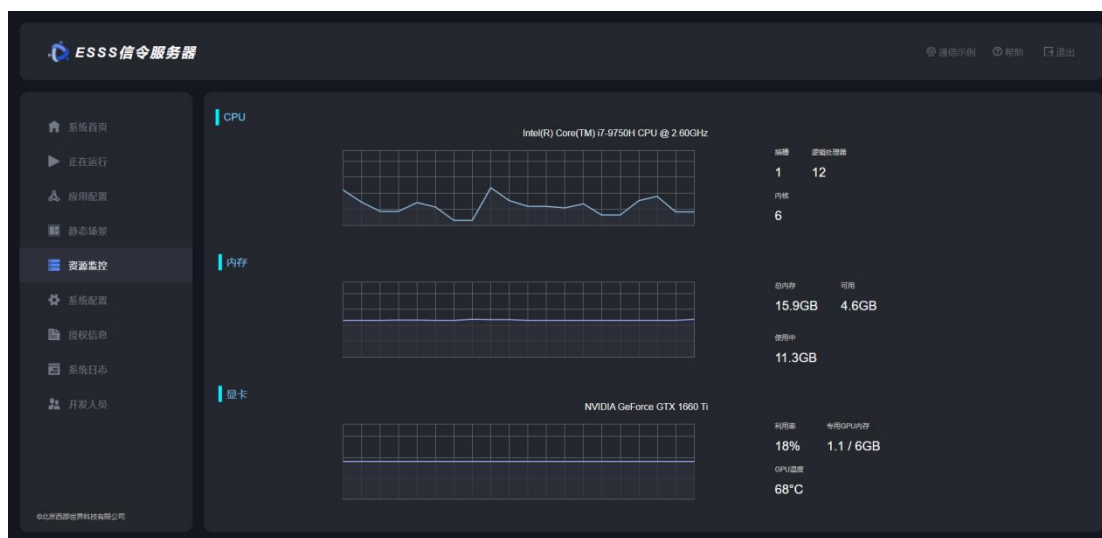




7. 资源监控

为了更好的进行资源监控管理、集中管控：根据服务器资源能力判定是否可启动实例，避免服务资源饱和崩溃；像素流应用、实例做到统一运维、统一管理，提高效率。延伸更多运营模式。

【资源监控】记录的是当前系统运行的实时变化，包含：CPU，内存和显卡的实时变化信息。

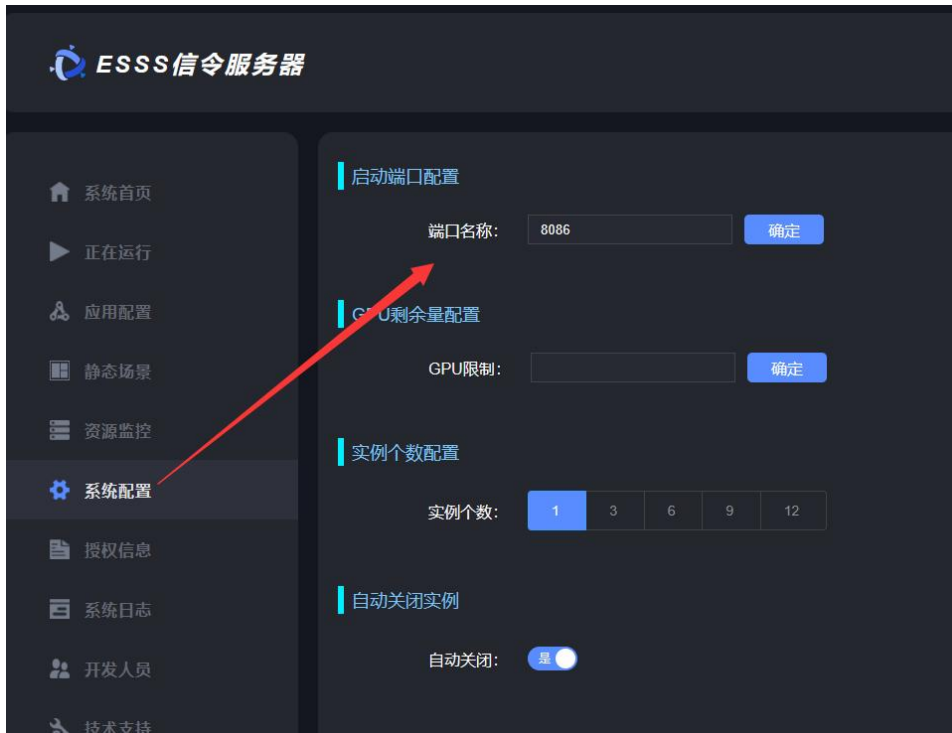


8. 系统配置

【系统配置】提供修改端口名称，最大实例个数，自动关闭实例的功能。

8.1. 启动端口配置

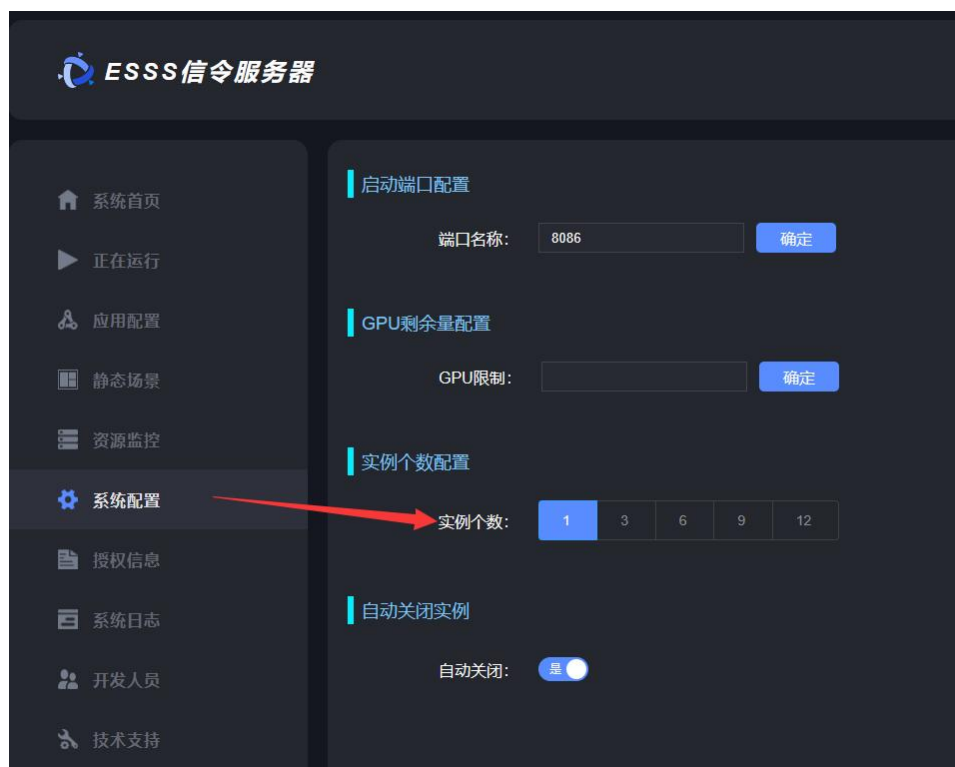
【通过启动端口配置】修改端口号需要重启服务生效。



8.2. 实例个数配置

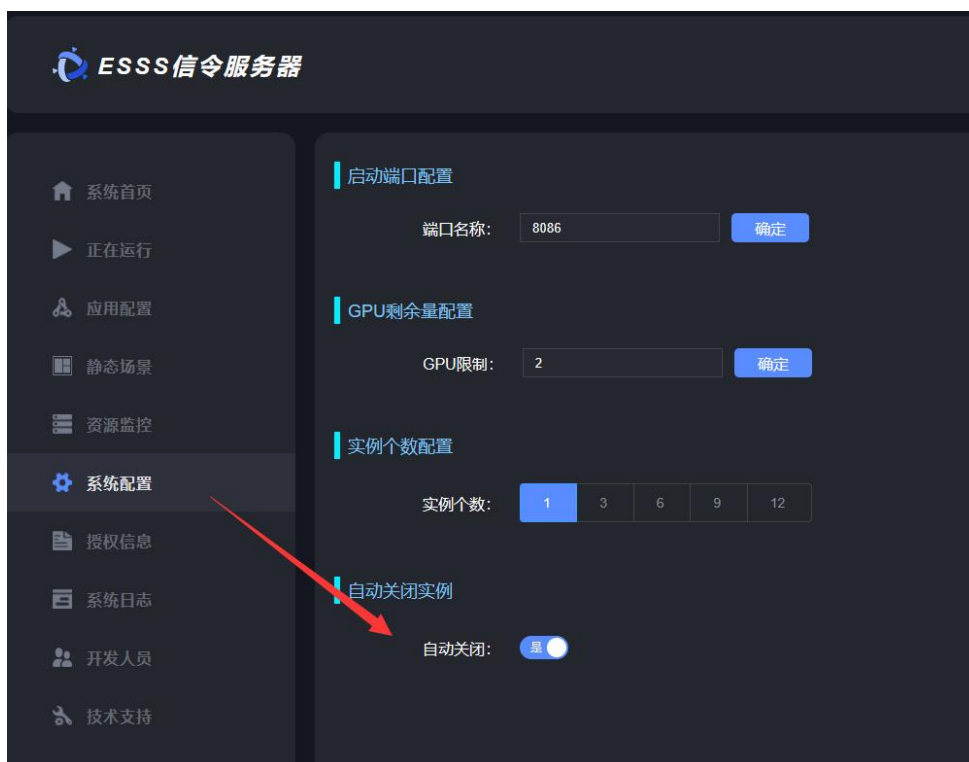
权限	云渲染实例个数
无授权	1 个实例
有授权	3、6、9、12 个实例

未授权状态下，实例个数默认为1，不可配置。授权状态下更改【实例个数配置】后需要重启服务生效。



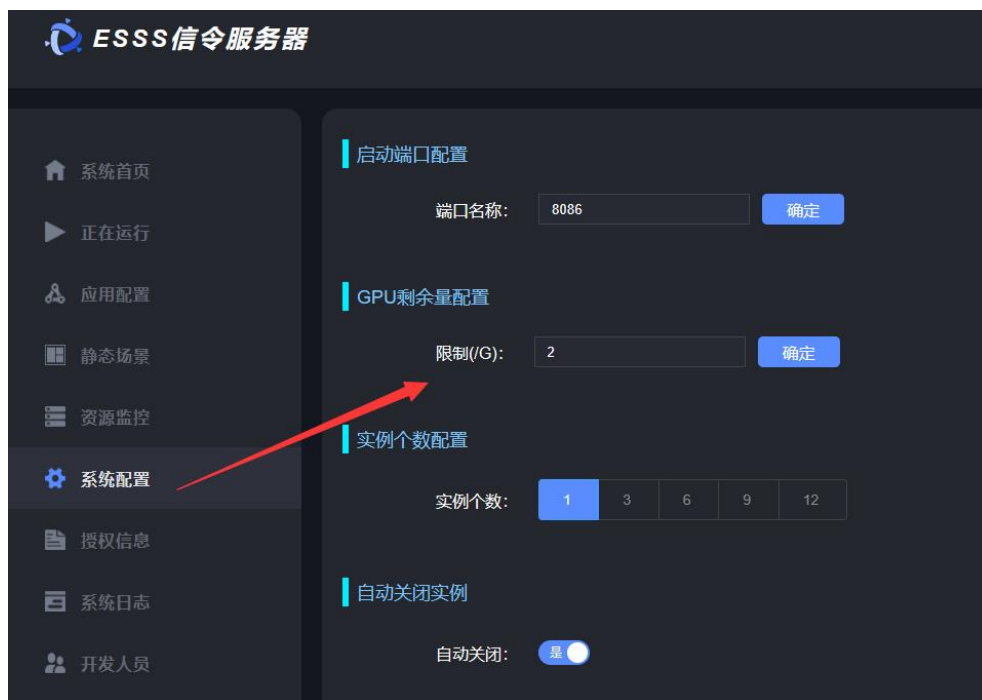
8.3. 自动关闭实例

【自动关闭实例】开启时，当没有连接时，实例自动关闭。



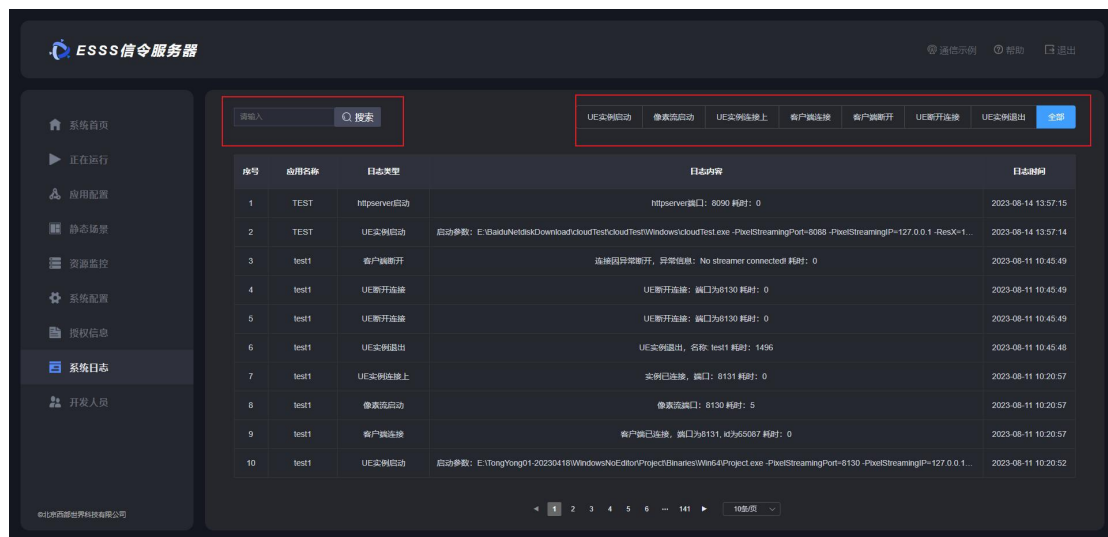
8.4. GPU 剩余量限制

【配置 GPU 余量】当启动 UE 程序时，当前机器所有 GPU 显存余量都低于设置的剩余量时，系统会报错。默认剩余量为 2G。



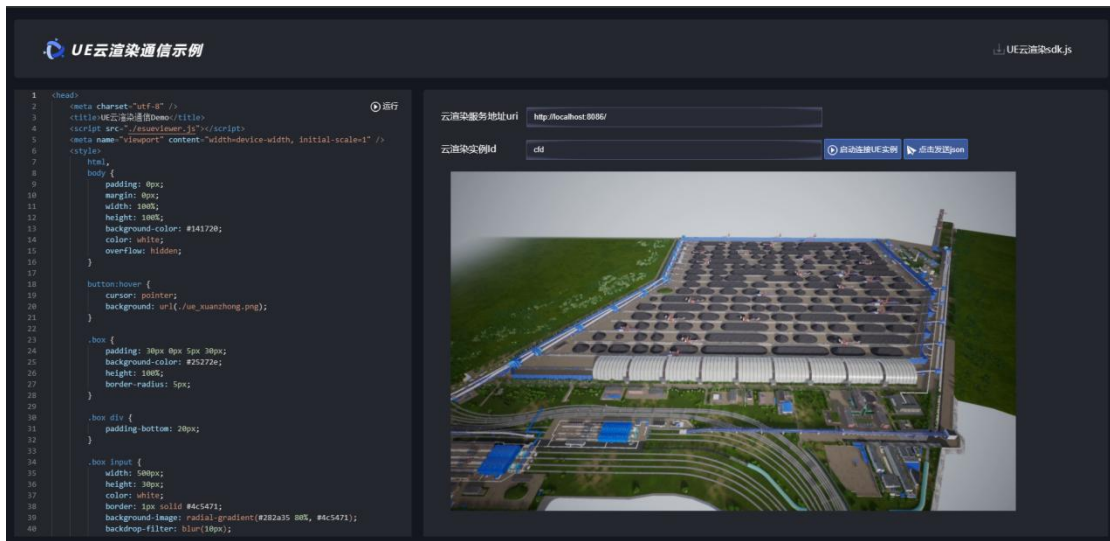
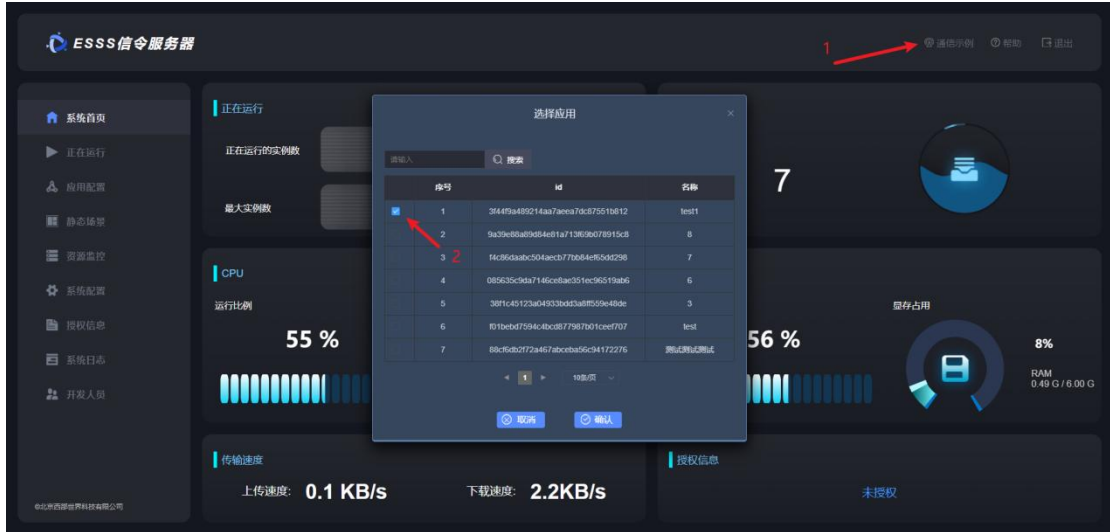
9. 系统日志

【系统日志】是对当前服务的日志信息的记录，包括：应用名称，日志类型，日志内容，日志时间。可以通过关键字进行模糊查询，以及通过按钮根据不同条件进行筛选。



10. UE 云渲染通信流程

点击【通信示例】，选择一个应用，点击确定跳转 UE 云渲染通信示例页面



1.1 引入 esueviewer.js 文件

```
<script src="./esueviewer.js"></script>
```

通信示例页面可通过右上角【UE 云渲染 sdk.js】下载

1.2 通过 websocket 服务建立通信关系

10.1.1. 拼接获取连接 UE 实例的请求地址

(1) 请求参数

获取启动实例的通信地址接口地址以及参数拼接	
uri	dom 元素的 id
signallingServerId	要启动的 UE 程序的 appid
width	云渲染分辨率的宽度
height	云渲染分辨率的高度

```
`${uri}instance/${signallingServerId}/${width}/${height}`
```

(2) 请求结果

①请求成功



②请求失败



10.1.2. 创建 UEViewer

构造函数参数	
domid	dom 元素的 id
{url:websocket 地址}	一个带有 url 属性的对象

```
const esUEViewer = new ESUEViewer("viewer", {url: resjson.server})
```

1.3 js 监听接收 UE 消息

监听事件含义	
ueevent	接收 ue pixel 发送消息
wserror	连接报错
wsclose	连接关闭

```
// (4) 网页 js 监听接收 UE 消息
esUEViewer.addEventListener('ueevent', (event) => {
    console.log("获取 UEEvent 消息:", event.detail)
})
// esUEViewer 连接失败内容
esUEViewer.addEventListener('wserror', (event) => {
    console.log('viewer ws error', event)
})
// esUEViewer 关闭事件
esUEViewer.addEventListener('wsclose', (event) => {
    console.log('viewer ws closed', event)
})
```

1.4 js 发送消息通信 UE

emitUIInteraction 函数参数	
msg	消息
callback	返回函数

```
esUEViewer.emitUIInteraction(json, () => {  
    console.log("发送完成")  
})
```