DTS Cloud For Linux 安装文档

- 一、安装包准备 🗄
- 二、初始化服务器 🗄
 - o 1. 创建 freedo 用户, 并授予 sudo 权限[⊕]
 - 2. 【可选,如果网络配置正常请跳过此步骤】网卡自动启动₺
 - 3. 处理系统防火墙並
 - 可选方法一:关闭防火墙服务 🗄
 - 可选方法二: 通过防火墙放行相关端口 🖞
 - o 4. 安装配置 nvidia 显卡驱动₺
 - 1. 安装依赖包 🗄
 - 2. 屏蔽自带驱动 🗄
 - 3. 显卡支持炎
 - 4. 重建 initramfs image む
 - 5. 重启む
 - 6. 停止 x windows 相关服务 [®]
 - 7. 上传下载后的驱动文件到服务器 🖞
 - 8. 为驱动文件添加执行权限 🖞
 - 9. 运行文件安装驱动 🗄
 - 10. 再次重启 🕹
 - 11. 重启之后执行命令`nvidia-smi`确定显卡驱动状态步
 - o 5.【可选】配置 tigervnc₺
 - 1. 安装配置 vncserver 服务 🗄
 - 2. 设置 freedo 用户的 vnc 密码 🕹
 - 3. 重启系统,确认 vnc 服务正常启动 🗄
 - 4. 使用 VNC Viewer 访问 ₺
- 三、上传安装包到服务器任意目录并解压步
 - 1. 创建部署目录 🗄
 - 。 2. 上传安装包 AirCityCloud_x86_0721.zip 到/deploy₺
 - 3. 解压安装包炎
- 四、申请授权 🖞
 - 1. 使用 PCIdentifier 工具生成机器码₺
 - 2. 提供机器码给飞渡,申请授权文件⊕
 - 3. 拿到授权文件, 重命名并复制到指定目录 ⊕
- 五、拷贝依赖库到指定目录。
 - 1. 拷贝依赖库到指定目录並
 - 1.1 X86架构む
 - 1.2 ARM64架构 t
- 六、启动集群管理服务 🗄
 - 1. 修改配置文件 AirCityCloud/CloudServer.conf
 - 2. 执行脚本, 启动集群管理服务 ⊕
 - 。 3. 【可选】配置为systemd服务,并开机启动⊕
 - 3. 访问实例管理页面[无渲染实例] ₺
- 七、启动渲染服务 🗄
 - 1. 修改配置文件 AirCityCloud/NodeService.conf, 使渲染节点加入到集群管理服务中 ⊕
 - 2. 执行脚本, 启动渲染实例 ⊕
 - o 3. 【可选】配置为systemd服务,并开机启动⊕
 - 4. 添加工程数据并访问锁
- 八、安装视频流中继服务 🖞
 - o 1. 安装 docker 服务₺
 - 2. 上传镜像到服务器₺
 - 3. 挂载镜像₺
 - 4. 编辑启动脚本, 然后启动服务⊕
 - 5. 配置 AirCityCloud/Config/TurnServer.conf 配置文件, 启用中继服务
 - 6. 重启服务使中继服务配置生效党

- 九、公网端口映射配置並
 - 1. 【联系网络管理人员配置端口】在出口路由器上映射`8087/tcp`和`3478/udp`端口到公网⊕
 - o 2. 根据端口映射配置, 修改AirCityCloud/Config/TurnServer.conf ⊕
 - o 3. 重启CloudServer服务使端口映射配置生效⊕
 - 4. 获取集群管理地址₺
 - 5. 使用上一步获取的映射后的地址访问集群管理页面。
- 十、【可选】配置Https访问⊕
 - 1. 申请证书, 推荐[腾讯云免费证书](https://console.cloud.tencent.com/ssl) ₺
 - 2. 申请完成之后,下载nginx证书₺
 - 3. 上传下载的证书到服务器任意目录,并解压₺
 - 4. 配置AirCityCloud/Config/Https.conf配置文件, 启用Https服务 ₺
 - o 5. 重启服务使Https配置生效₺
 - o 6. 使用https访问集群管理页面 ⊕
 - o 7. NodeService服务配置修改【所有节点服务配置都要修改】 ₺
 - 8. 重启NodeService服务₺
- 十一、【可选】KylinV10系统配置nfs⊕
 - 1、环境准备む
 - o 2、启动 NFS 服务 ⊕
 - 3、客户端挂载 NFS 共享目录,并配置开机自动挂载 ⊕
- 十二、常见问题与解决思路⊕
 - 1. 实例无法启动₺
 - 2. 无法访问页面 ₺
 - o 3. player 页面正常的情况下无法获取视频流 ₺
 - 问题描述: 卡在以下画面无法获取视频流步
 - 排查步骤: ₺
 - 4. 实例卡死、性能问题₺
 - 。 5. 中继服务无法访问问题⊕

PDF版本文档	下载地址
0721[最新版]	点击查看下载
0717	点击查看下载
0710	点击查看下载
0705	点击查看下载
0615	点击查看下载
0606	点击查看下载
0529	点击查看下载
0511	点击查看下载

实测环境

系统: Kylin Linux Advanced Server V10 (Sword)(SP2)

架构: X86_64

用户: freedo[具有sudo权限]

一、安装包准备

分类	系统架 构	组件	安装包名称	下载地址	备 注
显卡驱动	x86_64	NVIDIA 显卡驱动	NVIDIA-Linux-x86_64- 525.116.03.run	https://pan.baidu.com/s/1JEqonlVlgdveURtQ4ytaDQ? pwd=he8x	

分类	系统架 构	组件	安装包名称	下载地址	备 注
显卡驱动	ARM64	NVIDIA 显卡驱动	NVIDIA-Linux-aarch64- 525.116.04.run	https://pan.baidu.com/s/1Y20wYbfHwtonVYLfcZOVjA? pwd=haxr	
显卡驱动	x86_64	NVIDIA 显卡驱动 【Tesla系列专 用】	NVIDIA-Linux-x86_64- 535.54.03.run	https://pan.baidu.com/s/1LvBEcxrdI_ekzMSsxLNytg? pwd=bhku	
显卡驱动	ARM64	NVIDIA 显卡驱动 【Tesla系列专 用】	NVIDIA-Linux-aarch64- 535.54.03.run	https://pan.baidu.com/s/1u9A_PZ0x6I_Y_hb9u0DpwA? pwd=k9cj	
中继服务	x86_64	中继服务 coturn 镜像	coturn-latest-amd64- 20230510.tar.gz	https://installpackages.gbim.vip/docker_update/coturn- latest-amd64-20230510.tar.gz	
中继服务	ARM64	中继服务 coturn 镜像	coturn-latest-arm64- 20230511.tar.gz	https://installpackages.gbim.vip/docker_update/coturn- latest-arm64-20230511.tar.gz	
AirCityCloud	x86_64	AirCityCloud	AirCityCloud_x86_0721.zip	https://pan.baidu.com/s/1Zdajttji2-LQSM4y5yns7w? pwd=n11u	
AirCityCloud	ARM64	AirCityCloud	AirCityCloud_arm64_0721.zip	https://pan.baidu.com/s/1WbdmEiQ66R6Hk2V2PP5vAg? pwd=7nxt	

二、初始化服务器

1. 创建 freedo 用户, 并授予 sudo 权限

```
1 adduser -u 9988 freedo #建立用户名为 freedo 的一般用户
2 passwd freedo #为用户 freedo 设置密码
3 #Changing password for user freedo.
4 #New UNIX password: ← 输入密码(密码不会被显示)
   #Retype new UNIX password: ← 再次输入密码确认两次密码一致
5
   #passwd: all authentication tokens updated successfully. ← 密码设置成功
6
7
8 #添加freedo用户到wheel组
9
   #usermod -aG wheel freedo
10
11
   #配置freedo用户sudo免密
12 echo "freedo ALL=(ALL:ALL) NOPASSWD:ALL" | tee -a /etc/sudoers >/dev/null
```

【以下步骤都登录到 freedo 用户进行操作】

2. 【可选,如果网络配置正常请跳过此步骤】网卡自动启动

Kylin V10 Server SP2 安装成功之后,默认网卡是禁用状态,需要配置网卡自动启用。

```
1 # 查看网卡信息
```

2 sudo ifconfig

- 3
- # 根据获取的网卡信息,把以下命令中的ens192改为实际中的网卡名称 4
- 5 sudo nmcli con mod ens192 connection.autoconnect yes

3. 处理系统防火墙

选择下面的任意一种方法,处理系统自带 firewalld 防火墙

1 sudo systemctl stop firewalld

2 sudo systemctl disable firewalld

可选方法二:通过防火墙放行相关端口

```
1 # 允许vnc访问
2
    sudo firewall-cmd --zone=public --add-port=5900-5902/tcp --permanent
3
4
    # 允许Cloud页面访问,默认8087/tcp端口,根据实际情况修改
    sudo firewall-cmd --zone=public --add-port=8087/tcp --permanent
5
6
7
    # 渲染程序连接的端口
8
    sudo firewall-cmd --zone=public --add-port=8088/tcp --permanent
    # 实例管理服务使用的端口
9
    sudo firewall-cmd --zone=public --add-port=8089/tcp --permanent
10
11
12
    # 允许P2P视频流传输
13
14
    sudo firewall-cmd --zone=public --add-port=48100-48200/udp --permanent
15
16
    # 允许中继服务访问,默认3478/tcp 3478/udp端口,根据实际情况修改
17
    sudo firewall-cmd --zone=public --add-port=3478/tcp --permanent
18
19
    sudo firewall-cmd --zone=public --add-port=3478/udp --permanent
20
    # 重载配置, 然后查看配置状态
21
    sudo firewall-cmd --reload
22
   sudo firewall-cmd --list-port
23
```

4. 安装配置 nvidia 显卡驱动

驱动下载地址: https://www.nvidia.com/Download/Find.aspx?lang=en-us

若显卡为Tesla系列,如V100、T4、A10、A40等显卡,请选择Tesla系列专用驱动,普通驱动无法适用。

sudo dnf install elfutils-libelf-devel libglvnd libglvnd-devel vulkan-loader vulkan-loader-devel

2. 屏蔽自带驱动

- 1 sudo touch /etc/modprobe.d/blacklist-nvidia-nouveau.conf
- 2 sudo bash -c "cat > /etc/modprobe.d/blacklist-nvidia-nouveau.conf" << EOF</pre>
- 3 blacklist nouveau
- 4 options nouveau modeset=0
- 5 EOF

3. 显卡支持

1 sudo touch /etc/modprobe.d/nvidia.conf

```
2 sudo bash -c "cat > /etc/modprobe.d/nvidia.conf" << EOF
3 options nvidia NVreg_OpenRmEnableUnsupportedGpus=1
4 EOF
```

4. 重建 initramfs image

```
1 sudo mv /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bak
```

2 sudo dracut /boot/initramfs-\$(uname -r).img \$(uname -r)

5. 重启

sudo systemctl reboot

6. 停止 x windows 相关服务

sudo systemctl stop lightdm vncserver@\:1

7. 上传下载后的驱动文件到服务器

略

8. 为驱动文件添加执行权限

sudo chmod +x NVIDIA-Linux-x86_64-525.116.03.run

9. 运行文件安装驱动

sudo ./NVIDIA-Linux-x86_64-525.116.03.run -m=kernel-open

[freedo@kylinV10Sp2Server-01 ~]\$ sudo chmod +x NVIDIA-Linux-x86_64-525.116.03.run	
[freedo@kylinV10Sp2Server-01 ~]\$ sudo ./NVIDIA-Linux-x86_64-525.116.03.run -m=kernel-open	
Verifying archive integrity OK	
Uncompressing NVIDIA Accelerated Graphics Driver for Linux-x86_64 525.116.03	
□ 友法又本到当前Xshell箇目的全部会话	

	mibin Accelerated draphics britter for Einax Aco_of (SESTINOTOS)	
Building kernel modules		
	16%	



Install NVIDIA's 32-bit compatibility libraries? Yes No

NVIDIA Software Installer for Unix/Linux www.nvidia.com

注意: 最后一步,提示是否自动配置 X Window 的时候,请选择"yes",否则会导致 vnc 无法调用独立显卡,从而无法运行 AirCityExplorer

Would you like to run the myidia-moonfig utility to automatically update your X configuration five so that the MUDIAX driver will be used when you rest	tart X? Any pre-existing X configuration file will be backed up.
Yes	

10. 再次重启

sudo systemctl reboot

11. 重启之后执行命令 nvidia-smi 确定显卡驱动状态

[freedo Mon May	o@kylir y 810	V10Sp :18:5	2Server 6 2023	-01 ~]	\$ nvidi	ia-smi					
+	IA-SMI	525.1	16.03	Drive	r Versi	ion: 52	5.116.03	CUDA	Versio	n: 12.0	
GPU Fan 	Name Temp	Perf	Persis Pwr:Us	tence- age/Ca	M Bus· p 	-Id Mer	Disp.A nory-Usage	Vo] GPL 	latile J-Util	Uncorr. Compute MIG	ECC M. M.
====== 0 39% +	NVIDIA 44C	GeFo P8	rce 22W	off / 350W	-+ 0000 -+	90000:13 5MiB	3:00.0 Off / 24576MiB		0%	Defa	N/A ult N/A
+ Proco GPU	esses: GI	CI		 ID T	 ype f	rocess	name			GPU Mem	+ ory



5. 【可选】配置 tigervnc

1. 安装配置 vncserver 服务

- 1 # 安装tigervnc服务端
- 2 sudo dnf makecache
- 3 sudo dnf install tigervnc-server tigervnc-server-module
- 4
- 5 # 配置为系统服务
- 6 sudo cp /lib/systemd/system/vncserver@.service /etc/systemd/system/

7 sudo sed -i 's+<USER>+freedo+g' /etc/systemd/system/vncserver@.service

8 sudo systemctl daemon-reload

9 sudo systemctl enable vncserver@:1.service

2. 设置 freedo 用户的 vnc 密码

1 vncpasswd

- 2 #Password:
- 3 #Verify:
- 4 #Would you like to enter a view-only password (y/n)? n

3. 重启系统,确认 vnc 服务正常启动

[freedo@kylin	nV105p2Server-01 ~]\$ sudo systemct1 status vncserver@\:1.service
vncserver@.	:1.service - Remote desktop service (VNC)
Loaded: lo	paded (/etc/systemd/system/vncserver@.service; enabled; vendor preset: disabled)
Active: ac	ctive (running) since Mon 2023-05-08 10:18:05 CST; 9min ago
Process: 17	730 ExecStartPre=/bin/sh -c /usr/bin/vncserver -kill :1 > /dev/null 2>&1 : (code=exited, status=0/SUCCESS)
Process: 18	845 ExecStart=/usr/bin/vncserver -autokill :1 (code=exited, status=0/SUCCESS)
Main PID: 18	955 (Xvnc)
Tasks: 22	
Memory: 51	19.4M
CGroup: /s	system.slice/system-vncserver.slice/vncserver@:1.service
	-1855 /usr/bin/Xvnc :1 -auth /home/freedo/.Xauthority -desktop kylinV10Sp2Server-01:1 (freedo) -fp catalogue:/etc/X11/fontpath.d -geometry 1024x768 -pn -rfbauth /home/freedo/.vnc/passwd -rfbport 59>
	-2261 sh -c (/home/freedo/.vnc/xstartup; /usr/bin/vncserver -kill :1) >> '/home/freedo/.vnc/kylinV105p2Server-01:1.log' 2>81 &
	-2262 /usr/bin/mate-session
	-2265 dbus-launchexit-with-session /usr/bin/mate-session
	-2304 /usr/bin/dbus-daemonsyslogforkprint-pid 6print-address 8session
	-2310 /usr/libexec/dconf-service
	-2314 gnome-keyring-daemonstart
	-2318 /usr/libexec/mate-settings-daemon
	-2323 marco
	-2345 caja
	-2349 mate-volume-control-applet
	-2351 /usr/bin/python3 /usr/local/lib64/mate-indicators/libexec/reset_applet_position.py
	-2355 nm-applet
	-2361 ukui-screensaver-backend
	-2363 /usr/libexec/gvfsd
	-2373 /usr/libexec/gvfsd-fuse /home/freedo/.cache/gvfs -f -o big_writes
	-2374 /usr/libexec/geoclue-2.0/demos/agent
	-2497 /usr/libexec/imsettings-daemon
	-2512 /usr/libexec/at-spi-bus-launcher
	-2517 /usr/bin/dbus-daemonconfig-file=/usr/share/defaults/at-spi2/accessibility.confnoforkprint-address 3
	-2525 /usr/libexec/gvfsd-trashspawner :1.10 /org/gtk/gvfs/exec_spaw/0
الكووي	-2532 /usr/bin/pulseaudiostartlog-target=syslog
السوم	-2571 /usr/libexec/gvfs-udisks2-volume-monitor
	-2605 /usr/libexec/at-spi2-registryduse-gnome-session
	-2635 /usr/libexec/gvfs-goa-volume-monitor
	-2654 /usr/libexec/goa-daemon
	-2744 mate-panel
	-2802 /usr/bin/fcitx -D
	-2806 /usr/libexec/pulse/gsettings-helper
	=2811 /usc/libexec/pulse/gconf-beloec

4. 使用 VNC Viewer 访问

VNC Viewer 下载地址: https://www.realvnc.com/en/connect/download/viewer/windows/ 🛛

V2 VNC Viewer	🛿 KylinV10SP2Server - Properties — 🗆 🗙	- 🗆 X
ile View Help	General Options Expert	
by RealVNC Enter a VNC Se		> <u>Sign in</u> •
	VNC Server: 192.168 5:5901 Name: KylinV10SP2Server Labels To nest labels, separate names with a forward slash (/) Enter a label name, or press Down to apply existing labels	
	Security Encryption: Let VNC Server choose ✓ ✓ Authenticate using single sign-on (SSO) if possible ✓ Authenticate using a smartcard or certificate store if possible	
	Privacy	
	OK Cancel	
V2 VNC Viewer		- 🗆 X
by RealVNC Enter a VNC Se	rver address or search	Sign in ▼
KylinV10SP2Server	KylinV10SP2Server - VNC Viewer Authentication Authenticate to VNC Server 192.168.20.145::5901 (TCP) Enter VNC Server credentials (Hint: NOT your RealVNC account details) Username: Password: •••••• © Remember password OK Cancel	
	Stop	

三、上传安装包到服务器任意目录并解压

1. 创建部署目录

- 1 sudo mkdir /deploy
- 2 sudo chown -R freedo.freedo /deploy

2. 上传安装包 AirCityCloud_x86_0721.zip 到/deploy

3. **解压安装包**

unzip AirCityCloud_x86_0721.zip

四、申请授权

1. 使用 PCIdentifier 工具生成机器码

- 1 cd /deploy/AirCityCloud/
- 2 ./PCIdentifier

[fi _____01 AirCityCloud]\$./PCIdentifier 机器码: 666 _____9B

2. 提供机器码给飞渡,申请授权文件



3. 拿到授权文件, 重命名并复制到指定目录





五、拷贝依赖库到指定目录

1. 拷贝依赖库到指定目录

1.1 X86架构

1 cd /deploy/

- 2 cd AirCityCloud/dep/dll/
- 3 cp lib* ../../CloudRenderer/AirCityExplorer/Binaries/Linux/

1.2 ARM64**架构**

- 1 cd /deploy/
- 2 cd AirCityCloud/dep/dll/
- 3 cp lib* ../../CloudRenderer/AirCityExplorer/Binaries/LinuxAArch64/

六、启动集群管理服务

名称	详情
Cloud集群管理服务:	CloudServer
配置文件:	AirCityCloud/CloudServer.conf
启动脚本:	AirCityCloud/startCloudServer.sh
默认监听端口:	8087/tcp

1. 修改配置文件 AirCityCloud/CloudServer.conf

1. 修改 serverIP 变量为本机 IP 地址



2. 【可选】修改服务端口, 默认 8087/tcp

网页与API访问端口 playerPort="8087"

2. 执行脚本, 启动集群管理服务



3. 【可选】配置为systemd服务,并开机启动

- 1 cd /deploy/
- 2 cd AirCityCloud/
- 4 # 先停止脚本直接启动的服务
- 5 ./stopCloudServer.sh
- 6

3

- 7 # 执行脚本自动配置服务
- 8 cd dep/
- 9 ./makeService_CloudServer.sh
- 10
- 11 # 启动CloudServer服务
- 12 sudo systemctl start AirCityCloudServer.service
- 13
- 14 # 查看CloudServer服务状态
- 15 sudo systemctl status AirCityCloudServer.service

[freedo@	r-01 dep]\$ sudo systemctl start AirCityCloudServer.service
[freedc	ver-01 dep]\$ sudo systemctl status AirCityCloudServer.service
AirCityC	CloudServer.service - Freedo AirCityCloud Service
Loaded:	loaded (/etc/systemd/system/AirCityCloudServer.service; enabled; vendor preset: disabled)
Active:	active (running) since Thu 2023-05-25 15:17:31 CST; 4s ago
Process:	297931 ExecStart=/home/freedo/DTS/AirCityCloudPro/startCloudServer.sh (code=exited, status=0/SUCCESS)
Main PID:	297945 (CloudServer_lin)
Tasks:	10
Memory:	26.9M
CGroup:	/system.slice/AirCityCloudServer.service
	-297945 /home/freedo/DTS/AirCityCloudPro/CloudServer/CloudServer_linhttpRoot="/home/freedo/DTS/AirCityCloudPro/SDK"playerPort=8087license="/home/freedo/DTS/AirCityCloudPro/CloudServer/Lice>
5月 25 15:	17:2 r-01 startCloudServer.sh[297931]:44
5月 25 15:	17:28 + r-01 startCloudServer.sh[297931]:3
5月 25 15:	17:29 r-01 startCloudServer.sh[297931]:22
5月 25 15:	17:30 k r-01 startCloudServer.sh[297931]:11
5月 25 15:	17:31 ε r-01 startCloudServer.sh[297931]:00
5月 25 15:	17:31 / · · · · · · · · · · · · · · · · · ·
5月 25 15:	17:31 r-01 startCloudServer.sh[297931]: ^_^
5月 25 15:	17:31
5月 25 15:	17:31
5月 25 15:	17:31 .r-01 systemd[1]: Started Freedo AirCityCloud Service.
F.C. 1.01	

3. 访问实例管理页面[无渲染实例]

← C	▲ 不安全 192.100.20.1 ■:8087/samples/locale_zh/manage/	A" 20		
	A A A A A A A A A A A A A A A A A A A			

🔿 CloudMaster

实例管理: (0个实例)

实例标识符	运行状态	运行时长	渲染节点	分辨率	帧率	模式	连接	次数	缩放	限制	显卡	锁定	工程	操作

测试页面:视频流测试 API示例 实例管理接口 实时运行状态 样例集锦 Hello World 二三维坐标转换

Initializing...

Call GetStatus

"command":1,"authorization":null}
Result: {"command":-1004,"data":{"enableLog":true,"logLevel":"Debug","udpCustomPorts":true,"udpMinPort":50000,"udpMaxPort":65535,"pauseWhenIdle":false,"hideCrashDialogs":false,"resetWhenNotResponding":5,"hideSidebar":false}}

七、启动渲染服务

名称	详情
渲染服务:	NodeService
配置文件:	AirCityCloud/NodeService.conf
启动脚本:	AirCityCloud/startNodeService.sh

子节点只需要启动本 NodeService 服务,不需要启动 CloudServer 以及中继服务

1. 修改配置文件 AirCityCloud/NodeService.conf, 使渲染节点加入到集群管理服务中

- 1、 修改 serverIP 变量为集群管理服务的 IP 地址
- 2、 修改 serverPort 变量为集群管理服务的端口
- 【默认8087端口; 启用Https时, 默认8089端口】

<mark>#</mark>!/usr/bin/env bash # 渲染节点服务配置文件

服务器IP地址 serverIP="192.168.20.145"

服务器端口 serverPort="8087"

脚本所在目录, AircityCloud所在的目录 Aircity_Root=\$PWD

2. 执行脚本, 启动渲染实例

2 cd AirCityCloud/

3

4 ./startNodeService.sh

🔇 CloudMaster

实例管理: (4个实例)

实例标识符	运行状态	运行时长	這染节点	分辨率	帧率	模式	连接	次数	缩放	限制	显卡	锁定	工程	操作
2511176939984	Running	00:25:59	192.168.20.145	1920x937	25	1	0	3	Y	Y	0	N	/home/freedo/DTS/DTSProject/vtpkwjn/vtpkwjn.acp	设置参数取消锁定启动停止
2511176939996	Running	00:25:52	192.168.20.145	1920x1080	25	1	0	0	Y	Y	1	N	/home/freedo/DTS/DTSProject/vtpkwjn/vtpkwjn.acp	设置参数取消锁定启动停止
2511176940008	Running	00:25:46	192.168.20.145	1920×1080	25	1	0	0	Y	Y	0	N	/home/freedo/DTS/DTSProject/vtpkwjn/vtpkwjn.acp	设置参数取消锁定启动停止
2511176940020	Running	00:25:39	192.168.20.145	1920×1080	25	1	0	0	Y	Y	1	N	/home/freedo/DTS/DTSProject/vtpkwjn/vtpkwjn.acp	设置参数取消锁定启动停止
测试页面:视频流测试 API	ーー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・													

3. 【可选】配置为systemd服务,并开机启动

1 cd /deploy/ 2 cd AirCityCloud/ 3 4 # 先停止脚本直接启动的服务 5 ./stopNodeService.sh 6 7 # 执行脚本自动配置服务 8 cd dep/ 0 /makeCenvice_NodeCenvice

- 9 ./makeService_NodeService.sh
- 10 11 # 启动NodeService服务
- 12 sudo systemctl start AirCityNodeService.service
- 13
- 14 # 查看NodeService服务运行状态
- 15 sudo systemctl status AirCityNodeService.service

freedo(-01 AirCityCloudProj\$ sudo systemct1 status AirCityNodeService.service
AirCityN	odeService.service - Freedo AirCityCloud Service
Loaded:	loaded (/etc/systemd/system/AirCityNodeService.service; enabled; vendor preset: disabled)
Active:	active (running) since Thu 2023-05-25 15:13:48 CST; 8s ago
Process:	296738 ExecStart=/home/freedo/DTS/AirCityCloudPro/startNodeService.sh (code=exited, status=0/SUCCESS)
Main PID:	296761 (mono)
Tasks:	439
Memory:	2.3G
CGroup:	/system.slice/AirCityNodeService.service
	-296761 mono /home/freedo/DTS/AirCityCloudPro/CloudRenderer/RendererAgent/NodeService.exeip=192.168.20.145port=8089dataRoot="/home/freedo/DTS/AirCityCloudPro/CloudRenderer/RendererAgent/N
	-296782 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/AirCityExplorer/Binaries/Linux/AirCityExplorer-Linux-Shipping -id=2512210311074 -projectpath=/home/freedo/DTS/AirCityCloudPro/SDK/media/proje
	-296785 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/AirCityExplorer/Binaries/Linux/AirCityExplorer-Linux-Shipping -id=2512210311085 -projectpath=/home/freedo/DTS/AirCityCloudPro/SDK/media/proje
	-296788 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/AirCityExplorer/Binaries/Linux/AirCityExplorer-Linux-Shipping -id=2512210311096 -projectpath=/home/freedo/DTS/AirCityCloudPro/SDK/media/proje
	-296791 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/AirCityExplorer/Binaries/Linux/AirCityExplorer-Linux-Shipping -id=2512210311107 -projectpath=/home/freedo/DTS/AirCityCloudPro/SDK/media/proje
	-296933 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=zygoteno-zygote-sandboxno-sandboxlocales-dir-path=/home/freedo/DTS/AirCityCloudPro/Cloud
	-296935 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=zygoteno-zygote-sandboxno-sandboxlocales-dir-path=/home/freedo/DTS/AirCityCloudPro/Cloud
	-296941 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=zygoteno-sandboxlocales-dir-path=/home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Bina
	-296942 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=zygoteno-sandboxlocales-dir-path=/home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Bina
	-296945 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=zygoteno-zygote-sandboxno-sandboxlocales-dir-path=/home/freedo/DTS/AirCityCloudPro/Cloud
	-296946 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=zygoteno-zygote-sandboxno-sandboxlocales-dir-path=/home/freedo/DTS/AirCityCloudPro/Cloud
	-296947 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=zygoteno-sandboxlocales-dir-path=/home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Bina
	-296948 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=zygoteno-sandboxlocales-dir-path=/home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Bina
	-296973 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=gpu-processfield-trial-handle=11900335550845050427,18213647869669702984,131072no-sandbox
	-296974 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=gpu-processfield-trial-handle=1501349583957061839,5662202684715684194,131072no-sandboxoz
	-296999 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=gpu-processfield-trial-handle=12972911339298957415,7373245993841430247,131072no-sandboxo
	-297005 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=gpu-processfield-trial-handle=7149277571050211753,4606713364906240720,131072no-sandboxoz
	-297200 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=utilityutility-sub-type=network.mojom.NetworkServicetield-trial-handle=11900335550845050427
	-297201 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=utilityutility-sub-type=network.mojom.NetworkServicefield-trial-handle=1501349583957061839,
	-297230 /home/freedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=utilityutility-sub-type=network.mojom.NetworkServicefield-trial-handle=7149277571050211753,
	297231 /home/treedo/DTS/AirCityCloudPro/CloudRenderer/Engine/Binaries/Linux/CloudWebHelpertype=utilityutility-sub-type=network.mojom.NetworkServicetield-trial-handle=12972911339298957415
月 25 15:	13.48 Server-01 systemd[1]. Starting Freedo AirCityCloud Service
月 25 15:	3:48 b Server-of sudo[29(7)]: freedo : TTY=unknown : PWD=/home/freedo/DTS/AirCityCloudPro : USER=root : COMMAND=/usr/bin/chmod -R a+x ./CloudRenderer/RendererAgent/BatchFiles
月 25 15:	13:48
月 25 15:	3:48 kvliu
月 25 15:	13:48 / ~ ver-01 startNodeService.sh[296738]: 开始启动NodeService.
月 25 15:	13:48
月 25 15::	13:48 /
月 25 15::	13:48 er-01 systemd[1]: Started Freedo AirCityCloud Service.
	AirCityCloudPro]\$

4. 添加工程数据并访问

1. 上传工程数据到服务器上

2. 点击"设置参数", 打开实例设置界面

CloudMaster

实例管理: (2个实例)

实例标识符	运行状态	运行时长	渲染节点	分辨率	帧率	模式	连接	次数	缩放	限制	显卡	锁定	工程		操作	
2511308762465	Stopped	00:00:00	192.168.20.139	1920×1080	25	1	0	0	Y	Y	-1	N		设置参数	取消锁定 启动 停止	
2511308762477	Stopped	00:00:00	192.168.20.139	1920×1080	25	1	0	0	Y	Y	-1	N		设置参数	取消锁定启动停止	
测试页面 (在实例列表中选择	中实例, 可访问	回指定的实例,	如果没有选中实例,	则自动分配):												
视频流测试 AP	I示例 实例管	管理接口 实	时运行状态 样例集	瞎锦 Hello Wor	Id <u></u> ⊒∃	维坐标转	t/2									
Initializing ×																
connecting (ws://192.16 WebSocket Connected!	8.20.139:80)87/manage	r?category=Node	Service)			实例I	D: 2511	13087624	65						
Call GetStatus							I	呈:								
Result: {"command":-10	stomPor	ts":tru(e,"resetWhenNotRespor	nding":5,"hideSidebar":false}}				
							分辨	率: 192	0 x	1080	HD	FHD 2	!К 4K			
									维自适应	客户端分	辨率	□限制量	最大分辨率			
							视频模式	đ:	高画质	0 1	勾衡	🗌 流畅				
													高级参数 确定	取消		

3. 复制粘贴之前上传的 acp 工程的绝对路径到工程空白框中



实例管理: (2个实例)

实例标识符	运行状态	运行时长	渲染节点	分辨率	帧率	模式	连接	次数	缩放	限制	显卡	锁定	工程	操作
2511308762465	Stopped	00:00:00	192.168.20.139	1920x1080	25	1	0	0	Y	Y	-1	N		设置参数取消锁定启动停止
2511308762477	Stopped	00:00:00	192.168.20.139	1920x1080	25	1	0	0	Y	Y	-1	N		设置参数取消锁定启动停止

测试页面 (在实例列表中选中实例,可访问指定的实例,如果没有选中实例,则自动分配):

视频流测试	API示例	实例管理接口	实时运行状态	样例集锦	Hello World	二三维坐标转换

Initializing... connecting (ws://192.168.20.139:8087/manager?category=NodeService) ... WebSocket Connected!

Call GetStatus

{"command":1,"authorization":null} Result: {"command":-1004,"data":{"enableLog":true,"logLevel":"Debug","udpCustomPorts"

哈布市车	1949			
	实例参数	设置	x	
:tru(实例ID: 工程:	2511308762465 /deploy/ <u>DemoData/vtpkwin/vtpkwin.acp</u>		e,"resetWhenNotResponding":5,"hideSi
	分辨率:	1920 X 1080 HD FHD 2K 4K ☑< 三维自适应客户端分辨率 □ 限制最大分辨率		
	视频模式:	○ 高画质 ● 均衡 ○ 流畅		
		高级参数 确定 取消	i	

4. 点击确定, 然后本实例就会自动启动

🔵 CloudMaster

实例管理: (2个实例)

[实例标识符	运行状态	运行时长	渲染节点	分辨率	帧率	模式	连接	次数	缩放	限制	显卡	锁定	工程	操作
	2511308762465	Running	00:00:22	192.168.20.139	1920×1080	25	1	0	0	Y	Y	-1	Y	deploy/DemoData/vtpkwjn/vtpkwjnacp	设置参数取消锁定启动停止
	2511308762477	Stopped	00:00:00	192.168.20.139	1920×1080	25	1	0	0	Y	Y	-1	N		设置参数 取消锁定 启动 <mark>停止</mark>

测试页面 (在实例列表中选中实例,可访问指定的实例,如果没有选中实例,则自动分配):

视频流测试 API示例 实例管理接口 实时运行状态 样例集锦 Hello World 二三维坐标转换

Initializing... connecting (ws://192.168.20.139:8087/manager?category=NodeService) ... WebSocket Connected!

Call SetInstanceParams {"async":false,"staticInstance":{"videoMode":1,"encodeRateControl":"VBR","multipass":0,"encodeMaxQP":35,"keyframeInterval":250,"maxBitrate":35,"id":"2511308762465","resX":1920,"resY":1080,"project":"/deploy/DemoData/vtpkwjn/vtpkwjn.acp","adjustResolution":true,"lim Instance is starting, please wait... Result: {"command":104,"result":0} [操作成功] {"command":104,"result":0}

5. 点击"视频流测试",然后就可以跳转到视频流页面



八、安装视频流中继服务

在需要跨网段访问的cloud服务的环境下,就需要部署中继coturn服务。 只有在同网段测试的情况下,不需要此服务。

coturn 是一个基于 TURN (Traversal Using Relay NAT)协议实现的用于在 NAT 网络下进行实时音视频通信的服务器软件。在网络层中, NAT 路 由器会将源 IP 地址重写为自己的公网 IP 地址,而 TURN 服务器则可以通过中继数据包的方式,将经过 NAT 路由器的音视频数据转发到目标客 户端, 实现穿越 NAT 的效果, 从而保证实时音视频通信的顺畅和稳定。

coturn提供了一个开源的免费实现,支持以下特性:

- 1. IPv4和IPv6网络的支持。
- 2. 高可用性和负载均衡的支持。
- 3. 证书验证和加密传输的支持。
- 4. 远程管理和监控的支持。
- 5. WebRTC (Web Real-Time Communications) 的支持。

6. 简单可扩展的支持。

coturn适用于WebRTC、VoIP等网络通信应用,也可以作为一个代理服务器提供中间服务,增强网络连接稳定性与可靠性。

1. 安装 docker 服务

- 1 sudo dnf install docker
- sudo systemctl enable docker 2
- sudo systemctl start docker 3
- 4
- # 使freedo用户可运行docker 5
- sudo usermod -aG docker freedo 6
- 7 newgrp docker
- 8
- # 设置内核支持ipv4转发 9

11 sudo sysctl -p

L次元数据过期检查: 1:32:11 前,执行于 20 依赖关系解决。	l]\$ sudo dnf install docker 023年05月09日 星期二 07时18分30秒。				
Package	Architecture	Version	Repository		Size
安装: docker-engine	x86_64	18.09.0-202.ky10	ks10-sp2-iso		51 M
事务概要					
总下载: 51 M 安装大小: 216 M 确定吗? [y/N]: y 下载软件包: docker-engine-18.09.0-202.ky10.x86_64.rpm	ı		68	MB/s 51 MB	00:00
总计 运行事务检查 事务检查成功。 运行事务测试 事务测试成功。 运行事条			68	MB/s 51 MB	00:00
准备中 : 安装 : docker-engine-18.09.0-202.ky1 运行脚本: docker-engine-18.09.0-202.ky1 Created symlink /etc/systemd/system/multi	.0.x86_64 .0.x86_64 -user.target.wants/docker.service → /usr	/lib/systemd/system/docker.service.			1/1 1/1 1/1
/sbin/ldconfig: /usr/lib64/libLLVM-7.so 7	不是符号链接				
验证 : docker-engine-18.09.0-202.ky1	0.x86_64				1/1
已安装: docker-engine-18.09.0-202.ky10.x86_64					

2. 上传镜像到服务器

x86 架构镜像: https://installpackages.gbim.vip/docker_update/coturn-latest-amd64-20230510.tar.gz aarch64 架构镜像: https://installpackages.gbim.vip/docker_update/coturn-latest-arm64-20230511.tar.gz

3. 挂载镜像

- 1 tar zxvf coturn-latest-amd64-20230510.tar.gz
- 2 docker load < coturn-latest-amd64-20230510.tar

		ver-01	coturn]\$ ls	
coturn-latest-	-20230213	.tar.gz		
[]	بالتعموديندة	r-01	coturn]\$ tar zxvf coturn-latest-20230213.tar.gz	
coturn-latest-	-20230213	.tar		
		ver-01	<pre>coturn]\$ sudo docker load < coturn-latest-20230213.ta</pre>	r
4695cdfb426a:	Loading	layer [>]	84MB/84MB
46e2019b70bf:	Loading	layer [>]	49.25MB/49.25MB
6b66680faf51:	Loading	layer [>]	4.193MB/4.193MB
fe03f45141bf:	Loading	layer [>]	1.591MB/1.591MB
Loaded image:	coturn/c	oturn:1	atest	

4. 编辑启动脚本, 然后启动服务

```
默认使用3478/tcp 和 3478/udp 端口
如果要修改中继服务为其他端口,比如9999, 替换下面脚本中的
-p 3478:3478 \
-p 3478:3478/udp \
为
-p 9999:3478 \
```

-p 9999:3478/udp \

- 1 cd /deploy
- 2 mkdir coturn
- 3 cat > start_coturn.sh << "EOF"</pre>
- 4 docker run -d \setminus
- 5 --name freedo-coturn-01 \
- 6 --restart=always \
- 7 -e DETECT_EXTERNAL_IP=yes \
- 8 -e DETECT_RELAY_IP=yes \
- 9 -p 3478:3478 \
- 10 -p 3478:3478/udp \
- 11 -p 5349:5349 \
- 12 -p 5349:5349/udp \
- 13 -p 48100-48200:48100-48200/udp \
- 14 coturn/coturn:latest \

4 5						
15	-nlog-file=stdout \					
16	min-port=48100 \					
17	max-port=48200 \					
18	user=freedo:freedo \					
19	realm=feidu \					
20	lt-cred-mech					
21	EOF					
22						
23	# 授予脚本执行权限					
24	<pre>chmod +x start_coturn.sh</pre>					
25	_					
26	# 启动脚本					
27	./start coturn.sh					
28	_					
29	# 查看服务是否正常					
30	sudo docker ps					
<pre>> sudo dod >name f >restar > -p 3478; > -p 3478; > -p 3478; > -p 5349; > -p 5349; > -p 5349; > -p 5349; > -p 48106 > coturn/a > -nlog > coturn/a > -nlog >min-pc >max-pc > EOF [</pre>	<pre>cker run -d \ freedo-coturn-01 \ rt=always \ :3478 \ :3478 \ :3478 \ :3478 \ :5349 \ :5349 \ coturn:latest \ g-file=stdout \ port=48100 \ port=48200 ,</pre>					
CONTAINER	ID IMAGE COMMAND	CREATED	STATUS	PORTS		
	INAMES					

5. 配置 AirCityCloud/Config/TurnServer.conf 配置文件, 启用中继服务

```
#!/usr/bin/env bash
# 中继服务配置文件
# 是否使用中继服务连接配置,默认不使用
useTurnServer="yes"
# 中继服务访问端口
turnServerPort="3478"
# 当中继服务无法使用udp端口时,使用tcp端口,默认使用udp
useTCP="no"
# 中继服务连接配置
stunServer="stun:$serverIP:$turnServerPort"
if [ "$useTCP" = "yes" ]; then
   turnProtocol="tcp"
   turnServer="turn:$serverIP:$turnServerPort?transport=$turnProtocol"
else
   turnServer="turn:$serverIP:$turnServerPort"
fi
turnUsername="freedo"
turnPassword="freedo"
# 视频流中继方式
iceTransportPolicy="relay"
```

- 1. 修改turnServerPort变量为上一步启动的中继服务的端口,默认3478
- 2. 修改useTurnServer变量, yes表示启用中继服务, no表示不启用
- 3. 修改useTCP变量,默认使用udp协议进行数据转发,yes表示使用TCP协议

6. 重启服务使中继服务配置生效

0:48100-48200->48100-48200/udp freedo-coturn-01

sudo systemctl restart AirCityCloudServer.service

九、公网端口映射配置

1. 【联系网络管理人员配置端口】在出口路由器上映射 8087/tcp 和 3478/udp 端口到公网

8087/tcp 为 player页面访问端口,保证页面和接口的正常访问 3478/udp 为 coturn中继服务访问端口,保证视频流正常传输

本文档测试环境映射配置

内网IP	内网端口	映射IP	映射端口
192.168.20.145	8087/tcp	43.227.255.154	20535/tcp
192.168.20.145	3478/udp	43.227.255.154	13489/udp

2. 根据端口映射配置,修改AirCityCloud/Config/TurnServer.conf

usoTCP-"no"	
usercr= no	
# 中继服务连接配置	
stunServer="stun:\$serverIP:\$turnServerPort"	
if ["\$useTCP" = "yes"]; then	
turnProtocol="tcp"	
<pre>turnServer="turn:\$serverIP:\$turnServerPort?transport=\$turnProtocol"</pre>	
else	
turnServer="turn:\$serverIP:\$turnServerPort"	
fi	
turnUsername="freedo"	
turnPassword="freedo"	
# 视频流中继方式	
iceTransportPolicy="relay"	
# 遗夕和;p映射	
# 只有在开启中继服务之后,在需要的情况下才设置useManningServer	
useManningServer="ves"	
ischappingschver - yes	
# 指定端口映射服务器的IP地址或域名	
mappingServerIP="43.227.255.154"	
# 指定中继服务的映射端口	
portTurnMapping="13489"	
# playerPort的映射端口	
playerPortMapping="20535"	
changecont_mappingServer(){ # \$1 住进来的ec.conf_ic的路径	
# \$1 很远来自知C_CONT.JS自知的意义。	ningSonvonTP" 1. thon
file solution in the second seco	tManning\" \n\ \ \"Path\"+g" \$1
fi	
}	

- 1. 修改 useMappingServer 为 yes
- 2. 修改 mappingServerIP 为映射后的IP
- 3. 修改 portTurnMapping 为 3478/udp 映射后的端口

4. 修改 playerPortMapping 为 8087/tcp 映射后的端口

3. 重启CloudServer服务使端口映射配置生效

sudo systemctl restart AirCityCloudServer.service

4. 获取集群管理地址

sudo systemctl status AirCityCloudServer.service

[freedo@kylinV105p2Server-01 dep]\$ sudo systemctl status AirCityCloudServer.service	
• AirCityCloudServer.service - Freedo AirCityCloud Service	
Loaded: loaded (/etc/systemd/system/AirCityCloudServer.service; enabled; vendor preset: disabled)	
Active: active (running) since Mon 2023-06-26 14:46:29 CST; 1s ago	
Process: 596915 ExecStart=/home/freedo/DTS/AirCityCloud/startCloudServer.sh (code=exited, status=0/SUCCESS)	
Main PID: 596941 (CloudServer_lin)	
Tasks: 10	
Memory: 27.2M	
CGroup: /system.slice/AirCityCloudServer.service	
🖵 596941 /home/freedo/DTS/AirCityCloud/CloudServer/CloudServer_linhttpRoot="/home/freedo/DTS/AirCityCloud/SDK"playerPort=8087license="/home/freedo/DTS/AirCityCl	loud/CloudServer/License.lic" >
6月 26 14:46:26 kylinV105p2Server-01 startCloudServer.sh[596915]:333	
6月 26 14:46:27 kylinV105p2Server-01 startCloudServer.sh[596915]:222	
6月 26 14:46:28 kylinV105p2Server-01 startCloudServer.sh[596915]:111	
6月 26 14:46:29 kylinV105p2Server-01 startCloudServer.sh[596915]:000	
6月 26 14:46:29 kylinV105p2Server-01 startCloudServer.sh[596915]:CloudServer已启动,可以继续	
6月 26 14:46:29 kylinV10Sp2Server-01 startCloudServer.sh[596915]: ^_^	
6月 26 14:46:29 kylinV10Sp2Server-01 startCloudServer.sh[596915]: ^_^ 服务启动成功!	
6月 26 14:46:29 kylinV10Sp2Server-01 startCloudServer.sh[596915]:「内网]请使用浏览器「推荐Chrome]打开网址: https://192.168.20.145:8087/samples/locale zh/manage/ 管理节点	
6月 26 14:46:29 kylinV10Sp2Server-01 startCloudServer.sh[596915]: ^_^ [公网]请使用浏览器[推荐Chrome]打开网址: https://43.227.255.154:20535/samples/locale_zh/manage/ 管理节点	
6月 26 14:46:29 kylinV105p2Server-01_systemd[1]: Started Freedo AirCityCloud Service.	

5. 使用上一步获取的映射后的地址访问集群管理页面

🔵 CloudMaster

渲染节点

ID	主机名称	IP地址	Nvidia显卡数	支持的实例数	已启动的实例数
N2513035618988	free hine	192.168.20.150	1	2	2
N2512806924852		192.168.20.145	2	4	3

实例管理

实例标识符	运行状态	运行时长	渲染节点	分辨率	帧率	模式	连接	次数	缩放	限制	显卡	锁定	工程
2512806925114	Running	00:10:40	192.168.20.145	1920x937	25	均衡	0	1	Y	N	0	Y	/home/freedo/DTS/DTSProject/新建
2512806925091	Running	00:10:40	192.168.20.145	1920x1080	25	均衡	0	0	Y	N	0	Y	/home/freedo/DTS/DTSProject/6005
2512806925103	Running	00:10:40	192.168.20.145	1920x1080	25	均衡	0	0	Y	N	1	Y	/home/freedo/DTS/DTSProject/qd-ac
2513035619754	Running	00:10:40	192.168.20.150	1920x1080	25	均衡	3	3	Y	N	-1	Y	/home/freedo/DTS/DTSProject/vtpkwj
2513035619765	Running	00:10:40	192.168.20.150	1920x1080	25	均衡	1	1	Y	Y	-1	Y	/home/freedo/DTS/DTSProject/vtpkwj
2512806925125	Stopped	00:00:00	192.168.20.145	2560x1440	25	均衡	0	0	N	Y	1	N	/home/freedo/DTS/AirCityCloud/SDK/media

测试页面(在实例列表中选中实例,可访问指定的实例,如果没有选中实例,则自动分配):

视频流测试 API示例 实例管理接口 实时运行状态 样例集锦 Hello World 二三维坐标转换

Initializing...

十、【可选】配置Https访问

1. 申请证书, 推荐腾讯云免费证书♂

2. 申请完成之后,下载nginx证书

我的证书											
全部	E式证书	上传证书	免费证书								
		 1. [子所 2. 自202: 	号用户必看】关于腾讯云: 8年3月21日起,个人账号?	SSL所有接口家 可申请50张免累	下载证书 服务器类型	Q	操作	×	ŧ C		×
		申请中			Tomcat (pfx格式)		帮助 下载		已过期		
		0 待提交 0			Tomcat (JKS格式)		帮助 下载		1		
					Apache (crt文件、key文件)		帮助 下载				
					Nginx (适用大部分场景) (pem文件	、crt文件、key文件)	帮助 下载				
				绑定域名	腾讯云宝塔面板(pem文件、crt文件、	key文件)	帮助 下载			T i	
		III 音注: 未明 有效期: 共	3 1 年,当前第 1 年		IIS (pfx文件) 其他 (pem文件、crt文件、key文件)		帮助 下载 帮助 下载				部署 下載 升级 更多 ▼
		ID.	与 1年, 当前第1年		根证书下载 (crt文件) 不会部署证书?	备案管家,正3	帮助 下载				部署下载升级 更多▼
		10: 音注: 不可P	 年,当前第 1 年			→************************************	10				劉除 升级 更多 ▼
		 ・ 新江: 末中へ ・ 末中へ ・ ・ ・ ・ ・・・・・・・・・・・・・・・・・・・・・			2024-03-27 07:59:59	•	0				部署 下载 升级 更多 ▼

3. 上传下载的证书到服务器任意目录,并解压

[free	r-01 cert]\$ tree -L 2 .	
	nginx	
	com_bundle.crt	
Ь	l.com_bundle.pem	
— b.	com.csr	
bi bi	3d.com.key	
b .	.com_nginx.zip	
1 directory, 5 fi	iles	

4. 配置AirCityCloud/Config/Https.conf配置文件, 启用Https服务

#!/usr/bin/env bash # HTTPS配置文件
是否使用HTTPS访问,默认不使用,如果使用HTTPS访问请对此配置中的各项指定正确的参数 useHttps="yes"
指定渲染程序连接的端口 streamerPort="8088" # 指定实例管理服务使用的端口 managerPort="8089"
#证书,如果提供了有效的证书参数,则提供https服务以代替http ssl_cert="/home/freedo/DTS/AirCityCloudPro/Config/cert/bim_nginx/lom_bundle.pem"
#私钥文件 ssl_key="/home/freedo/DTS/AirCityCloudPro/Config/cert/kcom_nginx,com.key"

#证书密码(可选,创建证书时设置了密码则此处就要设置对应密码) ssl_passphrase=""

5. 重启服务使Https配置生效

sudo systemctl restart AirCityCloudServer.service

6. 使用https访问集群管理页面



你的连接不是专用连接

攻击者可能试图从 192.168.20.145 窃取你的信息(例如, 密码、消息或信用卡)。

NET::ERR_CERT_COMMON_NAME_INVALID

隐藏高级

返回

A* 😘 💻 💶 💶 💷 🖳 🖉 💮 …







渲染节点

 \leftarrow

ID	主机名称	IP地址	Nvidia显卡数	支持的实例数	已启动的实例数
N2511308762192	kylinV10SP1Server-01	192.168.20.139	1	2	1
N2512210310871	kylinV10Sp2Server-01	192.168.20.145	2	4	1

实例管理

实例标识符	运行状态	运行时长	渲染节点	分辨率	帧率	模式	连接	次数	缩放	限制	显卡	锁定	工程	操作
2511308762477	Running	00:02:32	192.168.20.139	1207x543	40	均衡	1	1	Y	N	-1	Y	/deploy/DemoData/demo60/demo60.acp	设置参数 取消锁定 启 动停止
2511308762465	Stopped	00:00:00	192.168.20.139	2560x1440	39	均衡	0	0	Y	N	-1	N	/deploy/DemoData/vtpkwjn/vtpkwjn.acp	设置参数 取消锁定 启 动 停止
2512210311074	Stopped	00:00:00	192.168.20.145	1920×1080	25	均衡	0	0	Y	N	0	Y	/home/freedo/DTS/AirCityCloudPro/SDK/media/project/demo.acp	设置参数 取消锁定 启 动 停止
2512210311085	Stopped	00:00:00	192.168.20.145	1920×1080	25	均衡	0	0	Y	N	1	Y	/home/freedo/DTS/DTSProject/qd-acp/yj_test.acp	设置参数 取消锁定 启 动 停止
2512210311096	Stopped	00:00:00	192.168.20.145	1920×1080	25	均衡	0	0	Y	N	0	Y	/home/freedo/DTS/AirCityCloudPro/SDK/media/project/demo.acp	设置参数 取消锁定 启 动 停止
2512210311107	Running	00:21:15	192.168.20.145	1207x543	25	均衡	0	2	Y	N	1	Y	/home/freedo/DTS/AirCityCloudPro/SDK/media/project/demo.acp	设置参数 取消锁定 启 动 停止

测试页面(在实例列表中选中实例,可访问指定的实例,如果没有选中实例,则自动分配):

C ▲ 不安全 https://192.168.20.145:8087/samples/locale_zh/manage/

视频流测试 API示例 实例管理接口 实时运行状态 样例集锦 Hello World 二三维坐标转换

Initializing... connecting (wss://192.168.20.145:8087/manager?category=CloudMaster) ... WebSocket Connected!

Call GetStatus

{"command":1,"authorization":null}

7. NodeService服务配置修改【所有节点服务配置都要修改】



8. 重启NodeService服务

sudo systemctl restart AirCityNodeService.service

十一、【可选】KylinV10系统配置nfs

本文档中的用户的 uid 为 9988 gid 为 9988 共享目录和挂载目录均为 /deploy/shareData KylinV10Server 默认已安装 nfs-utils 和 rpcbind

以上参数根据系统实际情况进行灵活修改

1、环境准备

IP	功能
192.168.20.145	NFS 服务端
192.168.20.139	NFS 客户端

2、启动 NFS 服务

1. 配置防火墙【若 firewalld 已禁用,跳过本步骤】

1 sudo firewall-cmd --permanent --add-service=nfs

2 sudo firewall-cmd --permanent --add-service=mountd

3 sudo firewall-cmd --permanent --add-service=rpc-bind

4 sudo firewall-cmd --reload

2. 创建共享目录并授权

1 mkdir /deploy/shareData 2 chmod 777 /deploy/shareData

3. 修改 nfs 配置文件

- 1 # anonuid=9988, anongid=9988 若客户端用户的uid/gid有变化,请自行修改
- 2 sudo bash -c "cat >> /etc/exports" << "JIESHU"</pre>
- 3 /deploy/shareData 192.168.20.0/24(rw,sync,insecure,no_subtree_check,no_root_squash,anonuid=9988,anongid=9988)

4 JIESHU

4. 启动 nfs 服务,并配置开机启动

1 sudo systemctl start nfs.service

3、客户端挂载 NFS 共享目录,并配置开机自动挂载

1. 确认可正常访问共享

showmount -e 192.168.20.145

2. 配置开机自动挂载

1 mkdir /deploy/shareData
2
3 sudo bash -c "cat >> /etc/fstab" << "JIESHU"
4 192.168.20.145:/deploy/shareData /deploy/shareData nfs4 defaults 0 0
5 JIESHU</pre>

3. 挂载目录

1 sudo mount -a

2 sudo df -hT

4. 测试

1	# 服务端
2	<pre>cd /deploy/shareData</pre>
3	touch ThisIsServer
4	
5	# 客户端
6	<pre>cd /deploy/shareData</pre>
7	touch ThisIsClient

十二、常见问题与解决思路

1. 实例无法启动

- 1. 检查是否已申请授权,并重命名为 License.lic,并放置到指定目录
- 2. 检查是否使用拥有 sudo 权限的普通用户运行 CloudServer 和 NodeService
- 3. 检查是否安装 vulkan 相关 rpm 包
- 4. 集群环境下,若实例一直处于 starting 状态,尝试先停止实例,再启动实例

2. 无法访问页面

1. 检查服务是否正常启动检查命令 ps -ef|grep AirCityCloud



2. 检查系统防火墙是否关闭,或者放行 cloud api 的端口,默认 8087/tcp

5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -X DOCKER' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -X DOCKER-ISOLATION-STAGE-1' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -X DOCKER-ISOLATION-STAGE-1' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -F DOCKER-ISOLATION-STAGE-2' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -F DOCKER-ISOLATION-STAGE-2' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -F DOCKER-ISOLATION' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -F DOCKER-ISOLATION' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -X DOCKER-ISOLATION' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -X DOCKER-ISOLATION' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -t filter -X DOCKER-ISOLATION' failed: iptables: No chain/target/match by that name. 5月 09 09:46:48 kylinV105p2Server-01 firewalld[34937]: WARNING: COMMAND_FAILED: '/usr/sbin/iptables -w10 -D FORWARD -i docker0 -o

3. player 页面正常的情况下无法获取视频流

问题描述:卡在以下画面无法获取视频流



🛛 🛕 不安全 📔 192.168.20.139:8087/samples/locale_zh/player.html

[09:42:52.919] sdk version: 6.0.0511 [09:42:52.928] uid: 1 [09:42:52.929] host: 192.168.20.139:8087 [09:42:53.132] connecting with ws... [09:42:53.140] connected [09:42:53.141] process started [09:42:53.142] setting up... [09:42:53.147] channel created [09:42:53.154] __ice_connection: new [09:42:53.154] __ice_gathering: new [09:42:53.155] __ice_gathering: gathering [09:42:53.294] __ice_gathering: complete [09:42:53.306] received answer: 1 [09:42:53.309] __signaling: stable [09:42:53.309] track: video [09:42:53.341] __ice_connection: checking [09:42:53.373] candidate added [09:42:53.373] candidate added [09:43:14.423] __ice_connection: disconnected [09:43:14.424] 连接建立失败,请检查服务器端口设置

排查步骤:

 \leftarrow

С

1. 未使用中继服务的情况下, 请确认已完成系统防火墙的配置

迪 过 刈 览 器 孔	J廾 Webrtc 调试地址 chrome://webrtc-inter	ials/ ,
← C @ E	dge edge://webrtc-internals	A* ta 🗸 🖉 🛆 🖡
 Create Dump Read stats From: S Note: computed state 	Standardized (promise-based) getStats() API	I and do not show up in the getStats result.
		http://192.168.20.145:8087/samples/loc iid=2511176939984 [rid: 91, lid: 13,
http://192.168.20 iceCandidatePool ICE connection st Connection state: Signaling state: n ICE Candidate pa ► ICE candidate g	0.145:8087/samples/locale_zh/player.html?iid=2511176939984, ISize: 0 } : new ew => have-local-offer => stable ir: (not connected) grid	{ iceServer stun:192.168.20.145:3478, turn:192.168.20.145:3478?transport=tcppiceTransportPolicy: relay, bundlePolicy: ba
Time	Event	Stats Tables
2023/5/11 14:32:16	► transceiverAdded	Filter statistics by type including separate multiple values by `,`
2023/5/11 14:32:16	► createDataChannel	media-playout (kind=audio, id=AP)
2023/5/11 14:32:16	► createOffer	certificate (id=CFC8:2F:AF:D9:22:E4:7F:A5:02:80:3E:F2:B3:E4:B8:B9:F1:F2:F8:43:D5:E7:88:10:67:4E:2
2023/5/11 14:32:16	negotiationneeded	► data-channel (id=D13)
2023/5/11 14:32:16	 createOfferOnSuccess (type: "offer", 3 sections) 	Indound-rtp (kind=video, mid=0, ssrc=3838364922, id=1101v3838364922) neer connection (id=P)
2023/5/11 14:32:16	 setLocalDescription (type: "offer", 3 sections) 	transport (iceState=new, dtlsState=new, id=T01)
2023/5/11 14:32:16	setLocalDescriptionOnSuccess	
2023/5/11 14:32:16	► signalingstatechange	
2023/5/11 14:32:16	► transceiverModified	
2023/5/11 14:32:16	► icegatheringstatechange	

3. 确认中继服务正常启动

2023/5/11 14:32:16 ► icecandidateerror

2023/5/11 14:32:16 setRemoteDescriptionOnSuccess 2023/5/11 14:32:16 signalingstatechange 2023/5/11 14:32:16 FtransceiverModified

[root@kylinV105p2Server-01 ~]# docker ps									
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS				
	NAMES								
7f6cb19f6848	coturn/coturn:latest	"docker-entrypoint.s"	2 days ago	Up 4 seconds	0.0.0.0:3478->3478/tcp, 0.0.0.0:3478->3478/udp, 0.0.0.0:5349->5349/udp, 0.0.0.0:5349->5349/tcp, 0.0.0.				
0:48100-48200->4810	00-48200/udp freedo-co	turn-01							
F 101 11 1405 05	0.4 J II								

4. 确认从服务器本地可以访问映射之后的中继服务端口

2023/5/11 14:32:16 setRemoteDescription (type: "answer", 3 sections)

2023/5/11 14:32:16 • addlceCandidate(sdpMid: 0, sdpMLineIndex: 0, type: host)

	[fr _dugkyrinvio_plConver-01 coturn]\$ telnet 192.168.20.145 3478
	Trying 192.168.20.145
	Connected to 192.168.20.145.
	Escape character is '^]'.
	^]
	telnet> quit
	Connection closed.
_	

5. 确认客户机可以联通中继服务端口而不是被防火墙拦截

C:\Users\.......i>telnet 192.168.20.145 3478 正在连接192.168.20.145...无法打开到主机的连接。 在端口 3478: 连接失败

4. 实例卡死、性能问题

1. 使用 htop 命令排查 cpu, 查看各个 cpu 核心使用率

htop 命令系统没有自带,请单独安装。

wget https://update.cs2c.com.cn/NS/V10/V10SP2/os/adv/lic/base/x86_64/Packages/htop-3.0.5-4.ky10.x86_64.rpm 1 sudo rpm -ivh htop-3.0.5-4.ky10.x86_64.rpm 2 3

4 htop

0[1[2[3[Mem[Swp[PID_USE	R <u>PRI</u>	NI <u>VIR</u> J	20 26 35 39 111111111111111111111111111111111	9.9%] 4[.2%] 5[.4%] 6[0.1%] 7[.1%] 7[35. 20. 19. 33. 1100000000000000000000000000000000000	<pre>7%] 8[</pre>	1930 thr; 6 running : 4.41 4.44 4.13 ys, 04:14:04	39.0%] 12[25.8%] 13[34.6%] 14[27.7%] 15[49.7% 50.3% 41.3% 30.3%
74482 free 74869 free 74869 free 74869 free 74903 free 74903 free 74903 free 72015 free 72015 free 75492 free 75492 free 75495 free 75495 free 75497 free 75487 free 75487 free 75497 free 75499 free 74417 free 74417 free 74417 free 74417 free 74417 free 74417 free 74417 free 74530 free 2484 free 74522 free 74523 free 2484 free 74523 free 2484 free 74523 free 2486 free 2487 free	edo 20 edo 20 edo 20 edo 20 edo 20 edo 20 edo 23 edo 20 edo 20	0 17.16 0 16.96 0 14.36 0 13.96 0 2545M 3 17.16 3 16.96 0 33.76 0 33.76 0 16.96 0 14.36 0 14.36 0 14.36 0 14.36 0 13.96 0 13.96 0 13.96 0 13.96 0 11316 5 16.96 0 11316 5 16.96 0 11316 0 3173M 0 3173M 0 3173M	5482M 1224M R 1 4899M 814M R 1 2926M 963M R 1 2436M 1054M S 3 145M 71840 S 3 145M 11840 S 3 5482M 1224M S 1 228M 198M S 1 2842M 1224M S 1 4899M 814M R 1 228M 198M S 1 2436M 963M S 1 2926M 963M S 1 2926M 963M S 1 201M 136M S 2 2436M 1054M S 2 2436M 1054M S 2434M 136M S 234M 147M S 204M 124M S 2436M 164M S </td <td>45. 11.7 38 28. 10.4 28 28. 10.4 28 28. 10.4 28 52. 13 5.2 24 51.9 0.3 5 24.3 11.7 6 41.8 10.4 4 9.9 0.5 2 9.2 0.5 1 8.6 11.7 3 6.0 10.4 2 4.1 6.2 2 7.7 0.5 1 6.6 10.4 2 7.7 0.4 1 7.7 0.5 1 6.4 5.2 3 5.1 0.4 1 3.8 10.4 1 3.8 10.4 1 3.8 10.4 4 3.2 0.5 6 1.9 0.0 4 1.9 0.0 6</td> <td>8:39.61 /home/freedo/DTS/AirCityCloudPro/A 8:55.05 /home/freedo/DTS/AirCityCloudPro/A 4:30.25 /home/freedo/DTS/AirCityCloudPro/A 3:37.42 /home/freedo/DTS/AirCityCloudPro/A 5h03:34 /usr/bin/Xvnc :1 -auth /home/freed 6:07.69 /home/freedo/DTS/AirCityCloudPro/A 4:56.70 /home/freedo/DTS/AirCityCloudPro/A 2:06.14 /opt/google/chrome/chrometype=g 1:59.71 /opt/google/chrome/chrometype=g 3:03.33 /home/freedo/DTS/AirCityCloudPro/A 2:43.77 /home/freedo/DTS/AirCityCloudPro/A 2:38.53 /home/freedo/DTS/AirCityCloudPro/A 1:50.30 /opt/google/chrome/chrometype=r 2:48.70 /home/freedo/DTS/AirCityCloudPro/A 1:27.40 /opt/google/chrome/chrometype=r 3:04.23 /home/freedo/DTS/AirCityCloudPro/A 1:08.43 /opt/google/chrome/chrometype=r 3:04.23 /home/freedo/DTS/AirCityCloudPro/A 0:53.23 /home/freedo/DTS/AirCityCloudPro/A 0:53.25 /home/freedo/DTS/AirCityCloudPro/A 0:53.25 /home/freedo/DTS/AirCityCloudPro/A 0:33.25 /home/freedo/DTS/AirCityCloudPro/A 0:33.25 /home/freedo/DTS/AirCityCloudPro/A 0:04.76 htop 7:02.08 /usr/bin/pulseaudiostartlog-</td> <td><pre>inCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit</pre></td> <td>yExplorer/Binaries/Linux yExplorer/Binaries/Linux yExplorer/Binaries/Linux yExplorer/Binaries/Linux ktop kylinV105p2Server-(yExplorer/Binaries/Linux pad-handler-pid=0enal pad-handler-pid=0enal pad-handler-pid=0enal pad-handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux yExplorer/Binaries/Linux yExplorer/Binaries/Linux</td> <td><pre>x/AirCityExplorer-Linux-Shipping - x/AirCityExplorer-Linux-Shipping - x/AirCityExplorer-Linux-Shipping - al:1 (freedo) -fp catalogue:/etc/Y x/AirCityExplorer-Linux-Shipping - x/AirCityExplorer-Linux-Shipping - ble-crash-reporter=,change-stac ble-crash-reporter=,change-stac ble-crash-reporter=,change-stac ble-crash-reporter=,change-stac c/AirCityExplorer-Linux-Shipping - crash-reporter=,change-stack-g x/AirCityExplorer-Linux-Shipping - crash-reporter=,change-stack-g</pre></td> <td><pre>id=2511176940020 -projectpath=/hor id=2511176939984 -projectpath=/hor id=2511176940008 -projectpath=/hor id=2511176940008 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176939996 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor</pre></td>	45. 11.7 38 28. 10.4 28 28. 10.4 28 28. 10.4 28 52. 13 5.2 24 51.9 0.3 5 24.3 11.7 6 41.8 10.4 4 9.9 0.5 2 9.2 0.5 1 8.6 11.7 3 6.0 10.4 2 4.1 6.2 2 7.7 0.5 1 6.6 10.4 2 7.7 0.4 1 7.7 0.5 1 6.4 5.2 3 5.1 0.4 1 3.8 10.4 1 3.8 10.4 1 3.8 10.4 4 3.2 0.5 6 1.9 0.0 4 1.9 0.0 6	8:39.61 /home/freedo/DTS/AirCityCloudPro/A 8:55.05 /home/freedo/DTS/AirCityCloudPro/A 4:30.25 /home/freedo/DTS/AirCityCloudPro/A 3:37.42 /home/freedo/DTS/AirCityCloudPro/A 5h03:34 /usr/bin/Xvnc :1 -auth /home/freed 6:07.69 /home/freedo/DTS/AirCityCloudPro/A 4:56.70 /home/freedo/DTS/AirCityCloudPro/A 2:06.14 /opt/google/chrome/chrometype=g 1:59.71 /opt/google/chrome/chrometype=g 3:03.33 /home/freedo/DTS/AirCityCloudPro/A 2:43.77 /home/freedo/DTS/AirCityCloudPro/A 2:38.53 /home/freedo/DTS/AirCityCloudPro/A 1:50.30 /opt/google/chrome/chrometype=r 2:48.70 /home/freedo/DTS/AirCityCloudPro/A 1:27.40 /opt/google/chrome/chrometype=r 3:04.23 /home/freedo/DTS/AirCityCloudPro/A 1:08.43 /opt/google/chrome/chrometype=r 3:04.23 /home/freedo/DTS/AirCityCloudPro/A 0:53.23 /home/freedo/DTS/AirCityCloudPro/A 0:53.25 /home/freedo/DTS/AirCityCloudPro/A 0:53.25 /home/freedo/DTS/AirCityCloudPro/A 0:33.25 /home/freedo/DTS/AirCityCloudPro/A 0:33.25 /home/freedo/DTS/AirCityCloudPro/A 0:04.76 htop 7:02.08 /usr/bin/pulseaudiostartlog-	<pre>inCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit enderercrashpad irCityCloud/AirCit</pre>	yExplorer/Binaries/Linux yExplorer/Binaries/Linux yExplorer/Binaries/Linux yExplorer/Binaries/Linux ktop kylinV105p2Server-(yExplorer/Binaries/Linux pad-handler-pid=0enal pad-handler-pid=0enal pad-handler-pid=0enal pad-handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux -handler-pid=0enable- yExplorer/Binaries/Linux yExplorer/Binaries/Linux yExplorer/Binaries/Linux	<pre>x/AirCityExplorer-Linux-Shipping - x/AirCityExplorer-Linux-Shipping - x/AirCityExplorer-Linux-Shipping - al:1 (freedo) -fp catalogue:/etc/Y x/AirCityExplorer-Linux-Shipping - x/AirCityExplorer-Linux-Shipping - ble-crash-reporter=,change-stac ble-crash-reporter=,change-stac ble-crash-reporter=,change-stac ble-crash-reporter=,change-stac c/AirCityExplorer-Linux-Shipping - crash-reporter=,change-stack-g x/AirCityExplorer-Linux-Shipping - crash-reporter=,change-stack-g</pre>	<pre>id=2511176940020 -projectpath=/hor id=2511176939984 -projectpath=/hor id=2511176940008 -projectpath=/hor id=2511176940008 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176939996 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940020 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor yuard-on-fork=enabledisable-gpu- id=2511176940008 -projectpath=/hor</pre>
2. 使用 r Thu May	vidia-s	mi 命令	查看显卡道 3 2023	运行状态	annja natata smt				I
+ NVID] GPU Fan ====== 0 57% + 1 57%	IA-SMI Name Temp NVIDIA 59C NVIDIA 59C	525.1 Perf GeFo P2 GeFo P2	16.03 [Persiste Pwr:Usag rce 161W /	Oriver ence-M ge/Cap Off 350W Off 350W	Version: 525.116.03 Bus-Id Disp.A Memory-Usage 00000000:13.00 Off 6208MiB / 24576MiB 00000000:1B:00.0 Off 6464MiB / 24576MiB	CUDA Versio Volatile GPU-Util 21%	on: 12.0 Uncorr. ECC Compute M. MIG M. Befault N/A Default	显存 - GPU使用 率	
+					 		N/A	- -	
Proce GPU 	GI ID	CI ID	PIC	О Тур	pe Process name		GPU Memory Usage		
	N/A N/A N/A N/A N/A N/A	N/A N/A N/A N/A N/A N/A	1987 74383 74434 74597 74869 74921 75092	7 3 C+ 4 C+ 7 C+ 9 C+ 1 C+ 2 C+ 7	G /usr/libexec/Xorg +GtyExplorer-Linux +Gs/cef3.logsha +Gs/cef3.logsha +GtyExplorer-Linux +Gs/cef3.logsha +Gs/cef3.logsha	<pre>c-Shipping ared-files ared-files c-Shipping ared-files ared-files</pre>	4MiB 2082MiB 11MiB 11MiB 4063MiB 11MiB 11MiB		
	N/A N/A	N/A N/A	74482	2 C+ 3 C+	+GtyExplorer-Linux +GtyExplorer-Linux	k-Shipping k-Shipping	3928MiB 2521MiB		

看内存使用



5. 中继服务无法访问问题

有些服务器修改了内核参数 net.ipv4.ip_forward=0,关闭了IPv4转发,导致无法访问容器端口,从而无法访问容器部署的中继服务

- 修复方式
- 1. 修改/etc/sysctl.conf
- 1 sudo vi /etc/sysctl.conf
- 2 net.ipv4.ip_forward=1

2. 执行命令重载内核参数

sudo sysctl -p

3. 重启docker服务

sudo systemctl restart docker