

# GPU 云服务器

# 故障处理





【版权声明】

©2013-2025 腾讯云版权所有

本文档(含所有文字、数据、图片等内容)完整的著作权归腾讯云计算(北京)有限责任公司单独所有,未经腾讯云 事先明确书面许可,任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成 对腾讯云著作权的侵犯,腾讯云将依法采取措施追究法律责任。

【商标声明】

## 🕗 腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算(北京)有限责任公司及其关联公司所有。本文档涉及的第三方主体的 商标,依法由权利人所有。未经腾讯云及有关权利人书面许可,任何主体不得以任何方式对前述商标进行使用、复 制、修改、传播、抄录等行为,否则将构成对腾讯云及有关权利人商标权的侵犯,腾讯云将依法采取措施追究法律责 任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况,部分产品、服务的内容可能不时有所调整。 您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则, 腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务,及相应的技术售后服务,任何问题请联系 4009100100或 95716。



## 文档目录

#### 故障处理

GPU 实例异常处理指南 常见 Xid 事件的处理方法 GPU 实例相关日志收集 GPU 使用率显示 100% 控制台的 VNC 不可用

## 🔗 腾讯云

## 故障处理 GPU 实例异常处理指南

最近更新时间: 2025-03-07 11:44:14

本文档旨在提供用户有关如何处理 GPU 实例异常的指引,以帮助用户快速诊断和解决 GPU 实例相关的问题。以 下是一些排查和处理建议,可用于处理部分常见的 GPU 实例问题。

## 系统状态检测

对于 GPU 服务器建议用户维持较新的GPU驱动版本、禁用 nouveau 模块、打开 GPU 驱动内存常驻模式并配置 开机自启动。

对于 GPU 服务器,建议进行以下配置:

- 维持较新的、正确的 GPU 驱动版本。
- 禁用 nouveau 模块。
- 打开 GPU 驱动内存常驻模式并配置开机自启动。
- GPU 故障后,建议在官网控制台重启机器看看是否可以恢复。

### 检查 GPU 驱动

GPU 驱动下载注意事项:

- •从 NVIDIA 官方文档 官方高级驱动搜索 | NVIDIA 选择正确的 GPU 型号。
- 对于64位 Linux Os 建议直接选择 Linux 64-bit。
- 选择 NVIDIA 推荐/认证的驱动。

### 禁用 nouveau 模块

nouveau 是 NVIDIA 显卡的开源驱动程序,会与 NVIDIA 官方 GPU 驱动发生冲突,需要在系统下禁用 nouveau 模块。

以下命令没有任何输出表示 nouveau 模块已经禁用:

[root@localhost ~]# lsmod | grep -i nouveau

#### 以下输出表示 nouveau 模块没有禁用:

[root@localhost	~]#	lsmod	grep	p -i nouveau
nouveau		1662531	_ 0	
mxm_wmi		13021	_ 1	nouveau
wmi		19086	52	mxm_wmi,nouvea
i2c_algo_bit		13413	31	nouveau



video	24538	1 nouveau			
drm_kms_helper	176920	2 nouveau,vmwgfx			
ttm	99555	2 nouveau,vmwgfx			
drm	397988	<pre>6 ttm,drm_kms_helper,nouveau,vmwgfx</pre>			
i2c_core	63151	5			
drm,i2c_piix4,drm_kms_helper,i2c_algo_bit,nouveau					

#### 禁用 nouveau 模块的方法参考如下:

```
# CentOS 7
# 编辑或新建 blacklist-nouveau.conf 文件
[root@localhost ~]# vim /usr/lib/blacklist-nouveau.conf
blacklist nouveau
options nouveau modeset=0
# 执行如下命令并重启系统使内核生效
[root@localhost ~]# dracut -f
[root@localhost ~]# shutdown -ry 0
```

## 配置 GPU 驱动内存常驻模式

打开 GPU 驱动内存常驻模式可以减少 GPU 掉卡、GPU 带宽降低、GPU 温度监测不到等诸多问题。建议打开 GPU 驱动内存常驻模式并配置开机自启动。 GPU驱动内存常驻模式检查常用方法:

#### nvidia-smi 输出中 Persistence-M 状态为 On。

示例:



NVIDIA-SMI 535.161.08	Driver	Version: 535.161.08	CUDA Version: 12.2
GPU Name   Fan Temp Perf 	Persistence-M Pwr:Usage/Cap	Bus-Id Disp.A Memory-Usage	Volatile Uncorr. ECC   GPU-Util Compute M.     MIG M.
0 NVIDIA H800   N/A 27C P0 	<u>On</u> 71W / 700W	-   00000000:23:00.0 Off   0MiB / 81559MiB 	 0   0 Default   Disabled
1 NVIDIA H800   N/A 28C P0 	On 72W / 700W	00000000:33:00.0 Off       0MiB / 81559MiB 	0    0% Default    Disabled
2 NVIDIA H800   N/A 30C P0 	On 73W / 700W	00000000:43:00.0 Off   0MiB / 81559MiB 	0    0% Default     Disabled
3 NVIDIA H800   N/A 30C P0 	On 73W / 700W	00000000:63:00.0 Off   0MiB / 81559MiB 	0    0% Default     Disabled
4 NVIDIA H800   N/A 27C P0 	0n 73W / 700W	00000000:83:00.0 Off   0MiB / 81559MiB 	0    0% Default     Disabled
5 NVIDIA H800   N/A 27C P0 	0n 72W / 700W	00000000:A3:00.0 Off 0MiB / 81559MiB	0   0% Default     Disabled
6 NVIDIA H800   N/A 30C P0 	0n 74W / 700W	00000000:C3:00.0 Off   0MiB / 81559MiB 	0 0   0% Default     Disabled
7 NVIDIA H800   N/A 31C P0	0n 72W / 700W	00000000:E3:00.0 Off   0MiB / 81559MiB 	0   0% Default   Disabled
· •			
Processes:   GPU GI CI P   ID ID	ID Type Proce	ss name	GPU Memory   Usage

### nvidia-bug-report.log 中, Persistence Mode 为 Enabled。

示例:



#### GPU 驱动内存常驻模式开启方法如下:





- # vim /etc/rc.d/rc.local
- # 在文件中添加一行
- # nvidia-smi -pm 1
- # 赋予/etc/rc.d/rc.local文件可执行权限
- # chmod +x /etc/rc.d/rc.local
- # 重启系统进行验证

### 获取 GPU 序列号

获取实例所有的 GPU 序列号:

Serial Number	: 0324018045603	
Serial Number	: 0324018044864	
Serial Number	: 0324018027716	
Serial Number	: 0323918059881	

#### 获取指定 id 的 GPU 序列号:



## GPU 常见故障

#### GPU 不识别

GPU 识别状态检测时,首先要确保 lspci 命令识别所有 GPU,其次确保 nvidia-smi 命令识别所有 GPU。

#### Ispci 检查 GPU 识别情况

输入以下命令确保所有 GPU 识别正常,并且每个 GPU 末尾标识为(rev a1);若输出信息末尾为(rev ff),表示 GPU 异常。

```
lspci | grep -i nvidia
```

示例:

```
#如下命令表示识别到4个GPU,且末尾标识为(rev a1)的GPU状态正常; 41:00.0 GPU末尾标识
为(rev ff),表示该GPU状态异常。
~]# lspci | grep -i nvidia
3e:00.0 3D controller: NVIDIA Corporation Device 1db8 (rev a1)
```



3f:00.0 3D controller: NVIDIA Corporation Device 1db8 (rev a1) 40:00.0 3D controller: NVIDIA Corporation Device 1db8 (rev a1) 41:00.0 3D controller: NVIDIA Corporation Device 1db8 (rev ff)

#### nvidia-smi 检查 GPU 识别情况

输入以下命令检查 GPU 识别情况:

#### nvidia-smi

## 示例:用 nvidia-smi 命令看到的 GPU 卡数量与实际不一致,如下图所示,8 块 GPU 卡的实例用 nvidia-smi 命令看到只有 7 块 GPU 卡。

NVID	IA-SMI	470.8	82.01	Driver	Version: 470.82.01	CUDA Version:	11.4
GPU Fan	Name Temp	Perf	Persis Pwr:Us	tence-Ml age/Capl	Bus-Id Disp.A Memory-Usage	↓   Volatile Ur   GPU-Util ( 	ncorr. ECC   Compute M.   MIG M.
0 N⁄A	Tesla 41C	V100 P0	-SXM2 41W	0n   / 300W	00000000:1A:00.0 Off 0MiB / 32510MiB	   0% 	0   Default   N/A
1 N⁄A	Tesla 36C	V100- P0	-SXM2 42W	0n   / 300W   	00000000:1B:00.0 Off 0MiB / 32510MiB	+     0%	0   Default   N/A
2 N/A	Tesla 38C	V100 P0	-SXM2 44W	0n   / 300W	 00000000:3D:00.0 Off 0MiB / 32510MiB	-+     0% 	0   Default   N/A
3 N/A	Tesla 37C	V100 P0	-SXM2 42W	0n   / 300W   	00000000:3E:00.0 Off 0MiB / 32510MiB	   0% 	0   Default   N/A
4 N/A	Tesla 37C	V100 P0	-SXM2 41W	0n   / 300W	00000000:88:00.0 Off 0MiB / 32510MiB	-+     0% 	0   Default   N/A
5 N⁄A	Tesla 38C	V100 P0	-SXM2 40W	0n   / 300W	00000000:89:00.0 Off 0MiB / 32510MiB	-+     0% 	0   Default   N/A
6 N/A	Tesla 35C	V100 P0	-SXM2 40W	0n   / 300W   	00000000:B1:00.0 Off 0MiB / 32510MiB	-+     0% 	0   Default   N/A
· · · · · · · · · · · · · · · · · · ·							
Proc GPU	esses: GI ID	CI ID	P	ID Typ	pe Process name	( L	FPU Memory   Isage
No	runnin	g pro	cesses f	ound			   

() 说明:

建议重启实例尝试是否可以恢复;若重启后无法恢复,仍出现 GPU 状态异常,请联系平台为您排查处理。

## GPU 带宽异常



需要确保 GPU 当前带宽与额定带宽一致且为 x16。可以使用 lspci 命令或 nvidia-smi 命令进行 GPU 带宽检 查。

#### lspci 命令

查询额定带宽:

lspci -vvd 10de: | grep -i Lnkcap:

查询当前带宽:

lspci -vvd 10de: | grep -i Lnksta:

### nvidia-smi 命令检查

方法一 方法一						
nvidia-smi -q   grep -i -A 2 'Link width'						
#输出示例:						
[root@localhost ~]# nvidia-smi -q	grep -i -A 2 'Link width'					
Link Width						
Max	: 16x					
Current	: 16x					
Link Width						
Max	: 16x					
Current	: 16x					
方法二						

力法—

nvidia-smi --format=csv --querygpu=index,name,serial,gpu\_bus\_id,pcie.link.width.current

#### #输出示例:

```
[root@localhost ~]# nvidia-smi --format=csv --query-
gpu=index,name,serial,gpu_bus_id,pcie.link.width.current
```

index, name, serial, pci.bus\_id, pcie.link.width.current
0, Tesla P40, 1321020022261, 00000000:04:00.0, 16



1,	Tesla	P40,	1320220073456,	0000000:05:00.0,	16
2,	Tesla	P40,	1320220073723,	0000000:08:00.0,	16
3,	Tesla	P40,	1320220073383,	00000000:09:00.0,	16
4,	Tesla	P40,	1320220073482,	0000000:85:00.0,	16
5,	Tesla	P40,	1320220073313,	0000000:86:00.0,	16
6,	Tesla	P40,	1320220073379,	0000000:89:00.0,	16
7,	Tesla	P40,	1320220073579,	00000000:8A:00.0,	16

! 说明:

通常为硬件问题,请联系平台为您排查处理。

## GPU retired pages 计数检查

#### NVIDIA GPU ECC RMA 标准

NVIDIA GPU ECC RMA 标准详情可参见 NVIDIA 官网文档 NVIDIA GPU ECC RMA 标准。

#### ! 说明:

对于 GPU retired pages 计数,满足以下任一条件或 nvidia fieldiag 检测 fail 均可联系平台进行 GPU 更换。

Retired Pages参数中:

- 30天内产生的 double bit ecc ≥5。
- 质保期内 double bit ecc ≥10。
- 质保期内double bit ecc+single bit ecc≥60。

### retired pages 查询方法





# 如输出Pending 的标志为No表示所有H 被软件程序调用,不会再影响程序运行; # Yes 表示有需要被屏蔽的ECC报错地址	ECC <b>报错地址空间已经被屏蔽,报错地址空间后续不会再</b> , <b>需要重启系统或重置</b> GPU <b>使其变为</b> No。
方法二	
# <b>该方法只能查看</b> retired pages <b>计数,</b> # <b>查询某一块</b> GPU <b>的</b> ECC <b>计数:</b> [root@localhost ~]# nvidia-smi	<b>无法查看</b> retired pages <b>是否已经被屏蔽</b> -g -i 0   grep -i 'bit ecc'
Single Bit ECC	: 0
Double Bit ECC	: 0
# 查看所有GPU的retired pages计数: [root@inspur ~]# nvidia-smi -q Single Bit ECC Double Bit ECC Single Bit ECC Double Bit ECC	grep -i 'bit ecc' : 0 : 0 : 1 : 0

#### 方法三

# 该方法可以查看所有retired pages的产生时间,便于判断是否满足nvidia RMA标准。

# 该方法需要较新的GPU驱动版本支持,否则无法查看retired pages产生时间。

[root@localhost ~]# nvidia-smi -i <target gpu> --query-retiredpages=gpu\_name,gpu\_bus\_id,gpu\_serial,retired\_pages.cause,retired\_pages.t imestamp --format=csv

#### 处理建议

- 若 GPU retired pages 计数满足 NVIDIA RMA 标准则联系平台进行硬件更换。
- 若 GPU retired pages 计数不满足 NVIDIA RMA 标准,需要检查当前报错的地址空间是否被屏蔽,即
   Pending: No,否则可重启系统或重置 GPU 屏蔽报错地址后再次测试程序运行情况;屏蔽报错地址后程序仍
   受 ECC 报错影响,进行 fieldiag 检测,测试 FAIL 则联系平台进行 GPU 更换。
- 对于 Volatile 和 Aggregate 条目下出现的 GPU ECC 报错,可使用 nvidia-smi -p 0/1 进行清除。

## GPU ERR! 报错检查

GPU 运行过程中会出现风扇、功率等 ERR! 报错,可以通过检查 nvidia-smi 输出中是否包含 ERR! 报错判断。 **功率 ERR! 报错示例:** 



NVID	IA-SMI	450.1	.02.04 Driver	Version: 450.102.04	CUDA Version: 11.0	
GPU Fan	Name Temp	Perf	Persistence-M Pwr:Usage/Cap	Bus-Id Disp.A   Memory-Usage 	Volatile Uncorr. ECC     GPU-Util Compute M.     MIG M.	
0 N/A	Tesla 66C	T4 P0	ERR! On 70W	00000000:00:09.0 Off   11721MiB / 15109MiB 		
Proc   GPU 	esses: GI ID	CI ID	PID Ty	pe Process name	GPU Memory   Usage	

#### 风扇 ERR! 报错示例:

	/	۱		 t-
GPU Name   Fan Temp Per <sup>.</sup> 	Persistence–M Pwr:Usage/Cap	Bus-Id Disp.A   Memory-Usage 	Volatile Uncorr. ECC GPU-Util Compute M. MIG M.	
	0n 58W / 150W	00000000:00:09.0 Off   13195MiB / 22731MiB 	( 0% Default N/#	
1 NVIDIA A10  ERR! 58C P0	On 65W / 150W	00000000:00:0A.0 Off   13195MiB / 22731MiB 	81 0% Default N/#	+ -   -   -
2 NVIDIA A10 0% 57C P0	On 61W / 150W	00000000:00:0B.0 Off   7405MiB / 22731MiB 	0% Default N/#	+ )   :   .
3 NVIDIA A10   0% 61C P0	0n 66W / 150W	00000000:00:0C.0 Off   16409MiB / 22731MiB 	0% Default N/#	

也可以通过 nvidia-bug-report 日志中的 Fan Speed 或 Power Draw 字段是否为 Unknown Error 来判断。

#### () 说明:

升级 GPU 驱动至较新版本后,重启系统进行观察。若重启后问题仍存在请联系平台为您排查处理。

## Xid 错误

Xid 消息是 NVIDIA 驱动程序向操作系统的内核日志或事件日志打印的错误报告。Xid 消息表示发生了 GPU 错误,通常是由于驱动程序对 GPU 编程不正确或发送到 GPU 的命令被损坏。常见Xid事件及处理建议可以参见 常见 Xid 事件的处理方法。

若以上状态检测和常见故障无法解决问题,请联系平台工程师协助排查处理。

🔗 腾讯云



## 常见 Xid 事件的处理方法

最近更新时间: 2025-06-25 10:19:42

本文档介绍了 Xid 消息是什么,为用户提供常见 Xid 事件的含义解释与处理方法。

## 什么是 Xid 消息

Xid 消息是 NVIDIA 驱动程序向操作系统的内核日志或事件日志打印的错误报告。Xid 消息表示发生了 GPU 错误,通常是由于驱动程序对 GPU 编程不正确或发送到 GPU 的命令被损坏。GPU 硬件、NVIDIA 软件问题或者 用户应用程序出现问题时都有可能产生 Xid 消息。这些消息提供的诊断信息可供用户和 NVIDIA 使用,以帮助调试 报告的问题。

## 如何查询 Xid 报错信息

使用 GPU 实例时,可以通过执行以下命令,查看是否存在 Xid 相关报错,保存回显结果。

#### dmesg | grep -i xid

- 若检查项 GPU 节点上的 Xid 异常为空,说明无 Xid 消息。
- 若检查项 GPU 节点上的 Xid 异常不为空,您可按照以下不同 Xid 消息对应建议方法进行处理或者联系 平台进行支持。

## 常见 Xid 事件的处理方法

不同 Xid 事件的含义不同,下文以**用户是否可能自行解决**为标准,将常见的 Xid 错误及对应的处理建议分成两类进 行介绍,完整的 Xid 说明详见 NVIDIA XID 官网说明 。

## 尝试自行解决

当遇到以下 Xid 事件时,可通过以下处理建议尝试自行解决;若仍无法解决,您可以通过 在线支持 – 腾讯云 进行 反馈,腾讯云工程师 7 × 24 小时在线为您提供服务。

### XID 48 错误

XID 48: Double Bit ECC Error

当 GPU 发生不可纠正的错误时,会上报 Xid 48 事件。该错误也会同时反馈给用户的应用程序。通常需要重置 GPU 或重启CVM实例来清除这个错误。

**处理建议:**建议重启实例恢复;若重启后问题仍存在请联系平台为您排查处理。**若业务对 xid48的异常比较敏感,** 可以直接要求换卡。

#### XID 79 错误

XID 79: GPU has fallen off the bus



此错误一般是 GPU 驱动或硬件问题,用户感知 GPU 实例存在掉卡现象。 **处理建议:**建议重启实例恢复;若重启后问题仍存在请联系平台为您排查处理。

## XID 94 错误

XID 94: Contained ECC error

此错误表示 GPU 发生了包含的 ECC 错误, 涉及 GPU 卡的应用会停止。

**处理建议:**建议重新启动应用验证业务是否正常,若重新启动应用后异常建议重启实例恢复;若重启后问题仍存在请 联系平台为您排查处理。

## XID 95 错误

XID 95: Uncontained ECC error

此错误表示 GPU 发生了未包含的 ECC 错误,涉及 GPU 卡的应用会停止。 **处理建议:**建议重启实例恢复;若重启后问题仍存在请联系平台为您排查处理。

## XID 119 错误

XID 119: GSP RPC Timeout

此错误一般是 GPU 驱动触发 GPU 系统处理器(GSP )bug 导致异常。

处理建议:

1. 关闭 GSP。在新代次实例中,NVIDIA GPU 包含了 GSP 固件功能。GSP 旨在卸载 GPU 初始化以及其他管理任务。您可以参考以下步骤关闭 GSP: (更多相关详细信息,请参阅 NVIDIA 网站上的关闭GSP )

```
echo "options nvidia NVreg_EnableGpuFirmware=0" >
/etc/modprobe.d/nvidia-gsp.conf
cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bak
```

○ 如果是 centos/tlinux/redhat 系统:

dracut -f --kver \$(uname -r)

○ 如果是 ubuntu/debian 系统:

sudo update-initramfs -u

- 重启机器验证
- 检查是否禁用成功:查看相关值是否为 0,若为 0 则已禁用 GSP。

grep EnableGpuFirmware /proc/driver/nvidia/params



- 2. 若您不希望关闭 GSP, 您可以尝试通过切换驱动版本解决:
- 更新驱动至 535.129.03 及以上版本,新版本驱动对 GPU GSP 引发的 XID 119 错误问题进行了修复。
- 降级驱动版本至 470 的最新稳定版本 470.223.02,该版本驱动默认不会开启 GSP,不会触发 XID 119 错误。

## 联系平台处理

当遇到以下 Xid 错误时,建议您可以直接通过 在线支持 – 腾讯云 进行反馈,腾讯云工程师 7 × 24 小时在线为您 提供服务。

**处理建议:** 可参见 GPU实例相关日志收集 收集 GPU 日志,联系平台为您排查处理。

## XID 74 错误

#### XID 74: NVLink ERROR

此错误表示 GPU 检测到从 GPU 到另一个 GPU 或通过 NVLink 的 NVSwitch 的连接出现问题, 可能是 GPU 本身异常或互连的 GPU 卡异常。

## XID 92 错误

#### XID 92: High single-bit ECC error rate

此错误表示高的单比特 ECC Error,可能是硬件或驱动故障。



## GPU 实例相关日志收集

最近更新时间: 2024-06-04 15:48:31

本文档旨在提供有关如何收集 GPU 实例相关日志的指导,以帮助用户技术支持人员及平台分析和解决与 GPU 实 例相关的问题。以下是相关日志的收集指引,可用于有效地收集 GPU 实例日志。 收集到的日志可尝试自行进行分析处理,或提供给腾讯云工程师进行排查处理。

## 获取子机 dmesg、串口日志

用户子机内执行命令:

dmesg | grep -i nv

## 获取 NVIDIA GPU 日志

安装 GPU 驱动的系统下,root 用户任意目录下执行命令:

nvidia-bug-report.sh

执行命令后,当前目录下会生成日志压缩包:nvidia-bug-report.log.gz。



## GPU 使用率显示 100%

最近更新时间: 2024-10-28 14:52:53

## 现象描述

使用 GPU 计算型实例的过程中,在系统内部使用 nvidia-smi 查看 GPU 状态时,可能遇到没有运行任何使用 GPU 的应用,但 GPU 使用率显示100%的情况。如下图所示:

NVIDIA-SMI 375.51 Driver Version: 375.51	!
GPU Name Persistence-M  Bus-Id Disp.A   Volatile   Fan Temp Perf Pwr:Usage/Cap  Memory-Usage   GPU-Util	Uncorr. ECC   Compute M.
0 Tesla M40 24GB 0ff   0000:00:06.0 0ff     N/A 53C P0 68W / 250W   0MiB / 22939MiB   0%	θ   Default
1 Tesla M40 24GB 0ff   0000:00:07.0 0ff     N/A 47C P0 65W / 250W   0MiB / 22939MiB   100%	0   Default
* * * * * * * * * * * * * * * * * * * *	
Processes:   GPU PID Type Process name  ====================================	GPU Memory   Usage
No running processes found	

## 可能原因

实例加载 NVIDIA 驱动时, ECC Memory Scrubbing 机制造成。

### 解决思路

**在实例系统内执行** nvidia-smi -pm 1 命令,让 GPU Driver 进入 Persistence 模式。

### 处理步骤

1. 登录 GPU 计算型实例,执行以下命令:

nvidia-smi -pm 1

NVIDIA-SMI 375.51 Driver Version: 375.51 -----+-GPUNamePersistence-MBus-IdDisp.AVolatile Uncorr. ECCFanTempPerfPwr:Usage/CapMemory-UsageGPU-UtilCompute M. \_\_\_\_ 4-----0 Tesla M40 24GB 0ff | 0000:00:06.0 0ff | Θ N/A 53C P0 68W / 250W | 0MiB / 22939MiB | <u>0%</u> Default . . . . . . . -----1 Tesla M40 24GB 0ff | 0000:00:07.0 0ff | Θ N/A 47C P0 65W / 250W | 0MiB / 22939MiB | 100% Default | Processes: GPU Memory GPU PID Type Process name Usage No running processes found 

2. 执行以下命令,检查 GPU 使用率:

nvidia-smi

腾讯云

GPU 使用率正常,如下图所示:

[rootQVM_18_107_centos data]# nvidia-smi Tue Aug 29 15:31:39 2017											
I NVIDI	ia-smi	384.6			Drive	r Vei	rsion: 384	.66			:
l GPU l Fan	Name Temp	Perf	Persisto Pwr:Usao	ence-Mi ge/Capi	Bus-Id	Mem	Disp.A pry-Usage	¦ Volat ¦ GPU-U	ile til	Uncorr. ECC Compute M.	
¦ ¦ Ø ¦N∕A	Tesla 22C	P40 P8	10W /	0n   250W	0000000 0M	0:00  iB /	:03.0 Off 22912MiB	+	0%	0 Default	
¦ 1 ¦N∕A ⁺	Tesla 23C	P40 P8	9W /	0n   250W	0000000 0M	0:00  iB /	:06.0 Off 22912MiB	   	0% 	0 Default	+
+   Proce   GPU  ======   No r +	esses:	PID ===== 1 proc	Type Pro	 ocess r ======= und	name					GPU Memory Usage	+ +



## 控制台的 VNC 不可用

最近更新时间: 2025-06-04 16:37:02

## 现象描述

使用 VNC 登录 Windows 实例 或 使用 VNC 登录 Linux 实例 时,登录界面无法显示登录提示信息,例如 黑屏 或 仅显示 Windows Logo 或 Guest has not initialized the display。如下图所示:







## 可能原因

1. GPU 实例安装了图形驱动。

VNC 方式登录 GPU 实例时,默认访问 QEMU 模拟的 VGA 设备,获取操作系统的 Framebuffer,实现访问操作系统。安装了 GPU 图形驱动之后,Framebuffer 不再交由 VGA 处理,VNC 无法访问操作系统。

2. 由于其他原因导致操作系统启动失败,例如安装了和系统冲突的第三方软件等。

## 解决方式

- 1. 若实例有公网 IP,则可参见 使用 RDP 文件登录 Windows 实例 登录实例。
- 2. 针对安装图形驱动的 GPU 实例,可在该实例中手动安装 VNC Server,用户即可在本地通过 VNC Client 进行登录。

请自行获取 VNC Server/Client 安装包。

 如果确认是因为原因1导致,可通过TAT命令禁用显卡驱动,禁用后可使用控制台VNC方式登录实例,设置 显示设置>扩展这些显示器>仅在1上显示,再次使用TAT命令恢复显卡驱动即可。



TAT 命令可参见 使用 TAT 命令禁用和恢复显卡驱动 操作。

检查已安装的第三方软件,分析其可能导致无法通过 VNC 方式登录实例的原因。
 建议卸载该第三方软件,或重装系统。