Cloud GPU Service Troubleshooting

Copyright Notice

Trademark Notice

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Troubleshooting
GPU Instance Exception Handling Guide

Last updated: 2026-03-27 18:28:48

This document provides the guide on how to handle GPU instance exceptions, helping you diagnose and resolve problems related to GPU instances quickly. Below are some troubleshooting and resolution suggestions for addressing common GPU instance issues.

System Status Check

For GPU servers, it is recommended to maintain an updated GPU driver version, disable the nouveau module, enable the GPU driver persistence mode, and configure it to start automatically on startup.
For GPU servers, the following configurations are recommended:

- Maintain an up-to-date and appropriate GPU driver version.
- Disable the nouveau module.
- Enable GPU driver persistence mode and configure it to start automatically on startup.
- In case of GPU failure, it is recommended to restart the server through the official console to check if the issue is resolved.

Checking the GPU Driver

Must-knows for downloading GPU drivers:

- Select the correct GPU model from Official Advanced Driver Search | NVIDIA .
- For 64-bit Linux OS, it is recommended to select Linux 64-bit directly.
- Select a driver that is **recommended or certified** by NVIDIA.

Disabling the Nouveau Module

The nouveau driver is an open-source driver for NVIDIA GPUs, which conflicts with the official NVIDIA GPU driver. It should be disabled in the system.
If the following command produces no output, it indicates that the nouveau module has already been disabled:

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

The following output indicates that the nouveau module has not been disabled:

```
[root@localhost ~]# lsmod | grep -i nouveau
nouveau               1662531  0
mxm_wmi                 13021  1 nouveau
wmi                     19086  2 mxm_wmi,nouveau
i2c_algo_bit            13413  1 nouveau
video                   24538  1 nouveau
drm_kms_helper         176920  2 nouveau,vmwgfx
ttm                     99555  2 nouveau,vmwgfx
drm                    397988  6 ttm,drm_kms_helper,nouveau,vmwgfx
i2c_core                63151  5
drm,i2c_piix4,drm_kms_helper,i2c_algo_bit,nouveau
```

Methods to disable the nouveau module are as follows:

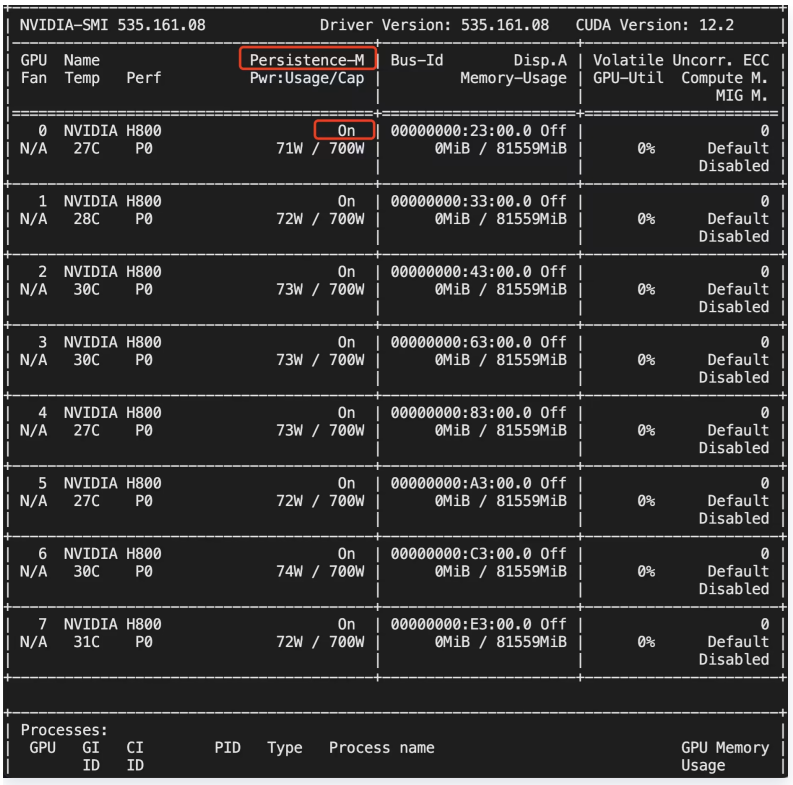
```
# CentOS 7
# Edit or create the blacklist-nouveau.conf file
[root@localhost ~]# vim /usr/lib/   /blacklist-nouveau.conf
blacklist nouveau
options nouveau modeset=0

# Execute the following commands and reboot the system to apply the
kernel changes
[root@localhost ~]# dracut -f
[root@localhost ~]# shutdown -ry 0
```

Configuring GPU Driver Persistence Mode

Enabling GPU driver persistence mode helps mitigate issues such as GPU disconnection, reduced GPU bandwidth, and inability to monitor GPU temperature. It is recommended to enable persistence mode and configure it to start automatically on startup.
Common methods to check GPU driver persistence mode:

In the output of nvidia-smi, the Persistence-M status is displayed as On.

Example:



```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.161.08             Driver Version: 535.161.08   CUDA Version: 12.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA H800                    On  | 00000000:23:00.0 Off |                    0 |
| N/A   27C    P0             71W / 700W |      0MiB / 81559MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA H800                    On  | 00000000:33:00.0 Off |                    0 |
| N/A   28C    P0             72W / 700W |      0MiB / 81559MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   2  NVIDIA H800                    On  | 00000000:43:00.0 Off |                    0 |
| N/A   30C    P0             73W / 700W |      0MiB / 81559MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   3  NVIDIA H800                    On  | 00000000:63:00.0 Off |                    0 |
| N/A   30C    P0             73W / 700W |      0MiB / 81559MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   4  NVIDIA H800                    On  | 00000000:83:00.0 Off |                    0 |
| N/A   27C    P0             73W / 700W |      0MiB / 81559MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   5  NVIDIA H800                    On  | 00000000:A3:00.0 Off |                    0 |
| N/A   27C    P0             72W / 700W |      0MiB / 81559MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   6  NVIDIA H800                    On  | 00000000:C3:00.0 Off |                    0 |
| N/A   30C    P0             74W / 700W |      0MiB / 81559MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   7  NVIDIA H800                    On  | 00000000:E3:00.0 Off |                    0 |
| N/A   31C    P0             72W / 700W |      0MiB / 81559MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
```

In the nvidia-bug-report.log, the Persistence Mode status is displayed as Enabled.

Example:

```
GPU 00000000:08:00.0
    Product Name                    : Tesla V100
    Product Brand                   : Tesla
    Display Mode                    : Enabled
    Display Active                  : Disabled
    Persistence Mode                : Enabled
```

The method to enable GPU driver persistence mode is as follows:

Method A

```
[root@localhost ~]# nvidia-smi -pm 1
```

Method B

```
# The following command is effective for newer versions of GPU drivers
[root@localhost ~]# nvidia-persistenced --persistence-mode
```

Configure autostart on startup:

```
# vim /etc/rc.d/rc.local
# Add a line to the file
# nvidia-smi -pm 1
# Grant execute permission to /etc/rc.d/rc.local
# chmod +x /etc/rc.d/rc.local
# Restart the system for verification
```

Retrieving GPU Serial Numbers

To retrieve the serial numbers of all GPUs in an instance:

```
# nvidia-smi -q | grep -i serial
    Serial Number                   : 0324018045603
    Serial Number                   : 0324018044864
    Serial Number                   : 0324018027716
    Serial Number                   : 0323918059881
```

To retrieve the serial number of a specified GPU ID:

```
# nvidia-smi -q -i 0 | grep -i serial
    Serial Number                   : 0324018045603
```

Common GPU Issues

GPU Not Recognized

When checking the GPU recognition status, first ensure that all GPUs are detected using the lspci command, and then verify their recognition with the nvidia-smi command.

Checking GPU Recognition Status with lspci

Run the following command to ensure all GPUs are recognized correctly. Each GPU should be marked with (rev a1) at the end. If the output shows (rev ff) at the end, it indicates a GPU exception.

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

Example:

```
# The following command output indicates that four GPUs are detected,
with those ending in (rev a1) being in a normal status. However, GPU
41:00.0, which ends in (rev ff), indicates an exception.
~]# 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)
```

Checking GPU Recognition Status with nvidia-smi

Run the following command to check GPU recognition status:

```
nvidia-smi
```

Example: The number of GPUs detected by the nvidia-smi command does not match the actual number of GPUs. As shown in the figure below, an instance with 8 GPUs displays only 7 GPUs when using the nvidia-smi command.

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.82.01    Driver Version: 470.82.01    CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:1A:00.0 Off |                    0 |
| N/A   41C    P0    41W / 300W |      0MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000000:1B:00.0 Off |                    0 |
| N/A   36C    P0    42W / 300W |      0MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  On   | 00000000:3D:00.0 Off |                    0 |
| N/A   38C    P0    44W / 300W |      0MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  On   | 00000000:3E:00.0 Off |                    0 |
| N/A   37C    P0    42W / 300W |      0MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   4  Tesla V100-SXM2...  On   | 00000000:88:00.0 Off |                    0 |
| N/A   37C    P0    41W / 300W |      0MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   5  Tesla V100-SXM2...  On   | 00000000:89:00.0 Off |                    0 |
| N/A   38C    P0    40W / 300W |      0MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   6  Tesla V100-SXM2...  On   | 00000000:B1:00.0 Off |                    0 |
| N/A   35C    P0    40W / 300W |      0MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Note:
It is recommended to restart the instance to check if the issue is resolved. If the issue persists after restarting CVM instances and the GPU remains in exceptional status, please contact the platform for further troubleshooting assistance.

GPU Bandwidth Exception

Ensure that the current GPU bandwidth matches the rated bandwidth and is set to x16. You can use the lspci command or the nvidia-smi command to check GPU bandwidth.

lspci Command

Check the rated bandwidth:

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

Check the current bandwidth:

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

Checking with the nvidia-smi Command

Method A

```
nvidia-smi -q | grep -i -A 2 'Link width'

#Sample output:
[root@localhost ~]# nvidia-smi -q | grep -i -A 2 'Link width'
            Link Width
                Max                 : 16x
                Current             : 16x
--
            Link Width
                Max                 : 16x
                Current             : 16x
--
```

Method B

```
nvidia-smi --format=csv --query-
gpu=index,name,serial,gpu_bus_id,pcie.link.width.current

#Sample output:
[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, 00000000:05:00.0, 16
2, Tesla P40, 1320220073723, 00000000:08:00.0, 16
3, Tesla P40, 1320220073383, 00000000:09:00.0, 16
4, Tesla P40, 1320220073482, 00000000:85:00.0, 16
5, Tesla P40, 1320220073313, 00000000:86:00.0, 16
6, Tesla P40, 1320220073379, 00000000:89:00.0, 16
7, Tesla P40, 1320220073579, 00000000:8A:00.0, 16
```

> **Note:**
> It is typically caused by hardware issues. Please contact the platform for further troubleshooting assistance.

Checking GPU Retired Pages Count

NVIDIA GPU ECC RMA Standards

For details on the NVIDIA GPU ECC RMA standards, see the official NVIDIA documentation NVIDIA GPU ECC RMA Standard.

> **Note:**
> **For GPU retired pages count, you can request GPU replacement by contacting the platform if any of the following conditions are met or if the NVIDIA fieldiag fails.**
> In the Retired Pages parameter:
> - Double-bit ECC errors ⩾ 5 occur within 30 days.
> - Double-bit ECC errors ⩾ 10 occur during the warranty period.
> - The sum of double-bit ECC and single-bit ECC errors ⩾ 60 during the warranty period.

Methods to Query Retired Pages

Method A

```
# Query the ECC count for a specific GPU by specifying the GPU ID:
[root@localhost ~]# nvidia-smi -i <target gpu> -q -d PAGE_RETIREMENT
    ...
    Retired pages
    Single Bit ECC             : 2
    Double Bit ECC             : 0
    Pending                    : No

# Query the ECC count for all GPUs:
[root@localhost ~]# nvidia-smi -q -d PAGE_RETIREMENT

# If the output shows Pending as No, it indicates that all ECC error
address spaces have been masked. These error address spaces will no
longer be called by software programs and will not affect program
operation in the future.
```

```
# If Yes is displayed, it indicates that there are ECC error addresses
that need to be masked. Restart the system or reset the GPU to change it
to No.
```

Method B

```
# This method only allows you to check the retired pages count but does
not indicate whether the retired pages have been masked.
# Query the ECC count for a specific GPU:
[root@localhost ~]# nvidia-smi -q -i 0 | grep -i 'bit ecc'
        Single Bit ECC              : 0
        Double Bit ECC              : 0

# View the retired pages count for all GPUs:
[root@inspur ~]# nvidia-smi -q  | grep -i 'bit ecc'
        Single Bit ECC              : 0
        Double Bit ECC              : 0
        Single Bit ECC              : 1
        Double Bit ECC              : 0
```

Method C

```
# This method allows you to check the generation time of all retired
pages, making it easier to determine whether the NVIDIA RMA standards
are met.
# This method requires a relatively new GPU driver version; otherwise,
the generation time of retired pages cannot be viewed.
[root@localhost ~]# nvidia-smi -i <target gpu> --query-retired-
pages=gpu_name,gpu_bus_id,gpu_serial,retired_pages.cause,retired_pages.t
imestamp --format=csv
```

Recommended Solution

- If the GPU retired pages count meets the NVIDIA RMA standards, contact the platform for hardware replacement.
- If the GPU retired pages count does not meet the NVIDIA RMA standards, check whether the current error address space is masked, indicated by Pending: No. Otherwise, restart the system or reset the GPU to mask the error addresses and retest the program. If the

program is still affected by ECC errors after masking the error addresses, perform a fieldiag test. If the test fails, contact the platform for GPU replacement.

- For GPU ECC errors under the Volatile and Aggregate entries, you can clear them using the command nvidia-smi -p 0/1.

GPU ERR! Error Check

During GPU operation, ERR! errors related to components such as fans and power may occur. These errors can be identified by checking the nvidia-smi output for any ERR! messages.

Power ERR! error example:

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.102.04   Driver Version: 450.102.04   CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            On   | 00000000:00:09.0 Off |                    0 |
| N/A   66C    P0   ERR! /  70W |  11721MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
```

Fan ERR! error example:

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.129.06   Driver Version: 470.129.06   CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A10          On   | 00000000:00:09.0 Off |                    0 |
|  0%   55C    P0    58W / 150W |  13195MiB / 22731MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA A10          On   | 00000000:00:0A.0 Off |                   81 |
| ERR!  58C    P0    65W / 150W |  13195MiB / 22731MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   2  NVIDIA A10          On   | 00000000:00:0B.0 Off |                    0 |
|  0%   57C    P0    61W / 150W |   7405MiB / 22731MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   3  NVIDIA A10          On   | 00000000:00:0C.0 Off |                    0 |
|  0%   61C    P0    66W / 150W |  16409MiB / 22731MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

You can also determine the issue by checking whether the Fan Speed or Power Draw fields in the nvidia-bug-report log show Unknown Error.

> **Note:**
> Upgrade the GPU driver to a newer version and restart the system to observe the results. If the issue persists after restarting, contact the platform for further troubleshooting assistance.

Xid Errors

XXid messages are error reports printed by NVIDIA drivers to the operating system’s kernel logs or event logs. These messages indicate a GPU error, often caused by incorrect GPU programming in the driver or corrupted commands sent to the GPU. For common Xid events and recommended solutions, see Troubleshooting Methods for Common Xid Events.
If the above status checks and common troubleshooting steps fail to resolve the issue, please contact the platform engineers for further troubleshooting assistance.

Handling Common Xid Events

Last updated: 2026-03-27 18:29:54

This document explains what Xid messages are and provides users with explanations of common Xid event meanings along with their handling methods.

What Are Xid Messages?

Xid messages are error reports printed by NVIDIA drivers in the kernel log or event log of the operating system. These messages indicate GPU errors, often caused by incorrect driver programming for the GPU or corrupted commands sent to the GPU. Xid messages can occur due to hardware issues, NVIDIA software problems, or user application errors. The diagnostic information provided by these messages can assist both users and NVIDIA in debugging the reported issues.

How to Check Xid Error Information?

When using a GPU instance, you can execute the following command to check for any Xid-related errors and save the result.

```
dmesg | grep -i xid
```

- If the Xid exceptions on the GPU node are empty during inspection, it indicates that no Xid messages are present.
- If the Xid exceptions on the GPU node are not empty during the inspection, you can follow the recommended solutions corresponding to different Xid messages or contact the platform for support.

Handling Common Xid Events

Different Xid events have varied implications. Below are common Xid errors and their corresponding solutions categorized based on **whether users can resolve them independently**. For a complete explanation of Xid messages, see NVIDIA XID official website explanation.

Attempting Self-Resolution

For the following Xid events, you can attempt to resolve the issue using the provided solutions. If the issue persists, you can contact Online Support - Tencent Cloud, where Tencent Cloud engineers are available 24/7 to assist you.

XID 48 Error

XID 48: Double Bit ECC Error
An Xid 48 event is reported when the GPU encounters an uncorrectable error. This error is also communicated to the user’s application. Typically, resolving this issue requires resetting the GPU or restarting CVM instances.
Solution: Restart CVM instances to resolve the issue. If the issue persists after restarting, please contact the platform for further troubleshooting assistance.

XID 79 Error

XID 79: GPU has fallen off the bus
This error is typically related to GPU driver or hardware issues, and users may notice GPU instances experiencing card disconnections.
Solution: Restart CVM instances to resolve the issue. If the issue persists after restarting, please contact the platform for further troubleshooting assistance.

XID 94 Error

XID 94: Contained ECC error
This error indicates a contained ECC error on the GPU, causing applications utilizing the GPU card to halt.
Solution: Restart the application to verify if the business operations return to normal. If the issue persists after restarting the application, restart CVM instances to resolve the error. If the issue persists after restarting, please contact the platform for further troubleshooting assistance.

XID 95 Error

XID 95: Uncontained ECC error
This error indicates an uncorrected ECC error on the GPU, causing applications utilizing the GPU card to halt.
Solution: Restart CVM instances to resolve the issue. If the issue persists after restarting, please contact the platform for further troubleshooting assistance.

XID 119 Error

XID 119: GSP RPC Timeout
This error is typically caused by a GPU driver triggering a bug in the GPU System Processor (GSP), leading to an exception.
Solution:
1. Disable GSP. In newer-generation instances, NVIDIA GPUs include GSP firmware features designed to offload GPU initialization and other management tasks. Follow the steps below

to disable GSP: (For more detailed information, see Disabling GSP on the NVIDIA website.)

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

- If you are using CentOS, Tlinux, or Red Hat systems:

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

- If you are using Ubuntu or Debian systems:

```
sudo update-initramfs -u
```

- Restart the machine to verify.
- Check if the disabling was successful: Verify whether the relevant value is 0. If it is 0, GSP has been successfully disabled.

```
grep EnableGpuFirmware /proc/driver/nvidia/params
```

2. If you prefer not to disable GSP, you can attempt to resolve the issue by switching the driver version:

- Update the driver to version 535.129.03 or later, as the new version includes fixes for XID 119 errors caused by GPU GSP.
- Downgrade the driver to the latest stable version of 470 (470.223.02), as this version does not enable GSP by default, avoiding XID 119 errors.

Contacting the Platform for Assistance

For the following Xid errors, it is recommended to directly report the issue through Online Support – Tencent Cloud. Tencent Cloud engineers are available 24/7 to assist you.
Solution: See GPU Instance-related Log Collection to collect GPU logs and contact the platform for further troubleshooting assistance.

XID 74 Error

XID 74: Nvlink ERROR

This error indicates that the GPU has detected an issue with the connection from one GPU to another GPU or through an NVSwitch via NVLink. The issue could stem from the GPU itself or

an interconnected GPU card.

XID 92 Error

XID 92: High single-bit ECC error rate

This error indicates a high single-bit ECC error, which may be caused by hardware or driver failure.

Collecting Log for GPU Instances

Last updated: 2025-02-24 15:32:25

This document guides you through collecting the logs related to GPU instances to assist users, technical support teams, and platforms in analyzing and resolving GPU instance-related issues. Below are the instructions for effectively collecting GPU instance logs. The collected logs can be analyzed and processed independently or provided to Tencent Cloud engineers for troubleshooting.

Retrieving Sub-instance dmesg and Serial Port Logs

Execute the command on the user instance:

```
dmesg | grep -i nv
```

Collecting NVIDIA GPU Logs

On a system with GPU drivers installed, execute the following command as the root user in any directory:

```
nvidia-bug-report.sh
```

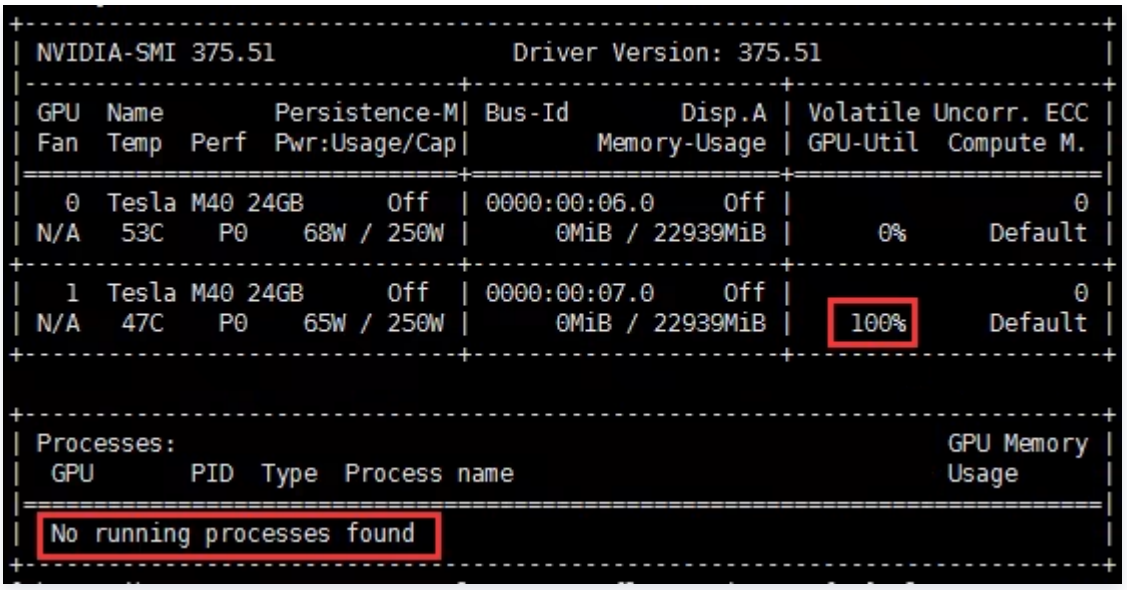
After the command is executed, a compressed log file named nvidia-bug-report.log.gz will be generated in the current directory.

GPU Usage Displaying as 100%

Last updated: 2025-02-24 15:32:58

Phenomenon Description

When using GPU compute-optimized instances, you may encounter a situation where the GPU usage displays as 100% in the system while checking GPU status using `nvidia-smi`, even though no GPU-related applications are running. An example is shown below:



Possible Reasons

This issue is caused by the ECC Memory Scrubbing mechanism when the instance loads the NVIDIA driver.

Problem-solving Ideas

Execute the command `nvidia-smi -pm 1` in the instance's system to enable the GPU driver to enter Persistence mode.

Processing Procedures

1. Log in to the GPU compute-optimized instance and execute the following commands:

```
nvidia-smi -pm 1
```

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.51                 Driver Version: 375.51                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla M40 24GB      Off  | 0000:00:06.0     Off |                    0 |
| N/A   53C    P0    68W / 250W |      0MiB / 22939MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla M40 24GB      Off  | 0000:00:07.0     Off |                    0 |
| N/A   47C    P0    65W / 250W |      0MiB / 22939MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
| No running processes found                                                  |
+-----------------------------------------------------------------------------+
```

2. Run the following command to check the GPU usage:

```
nvidia-smi
```

The GPU usage is normal, as shown in the following figure:

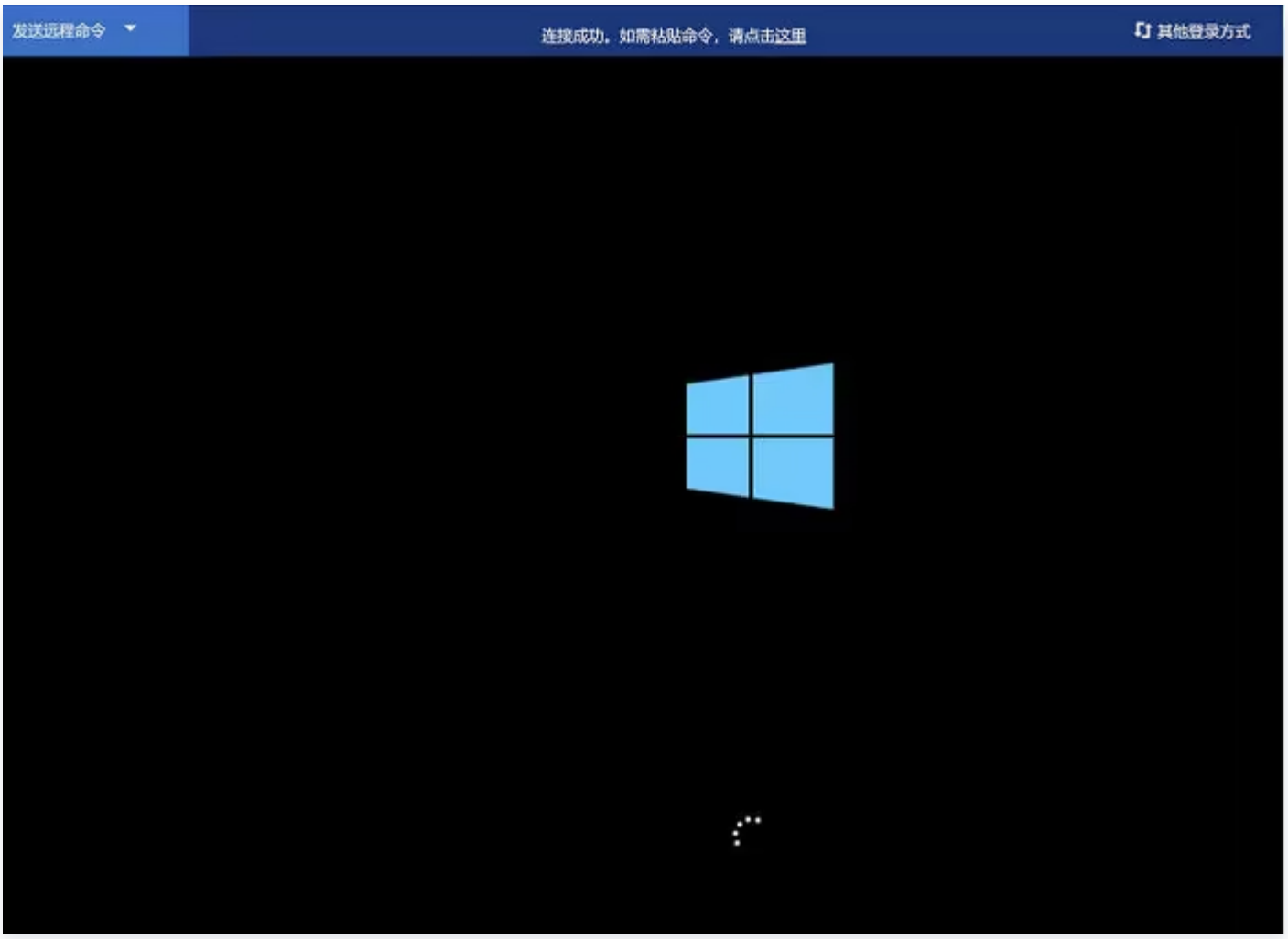
```
[root@VM_18_107_centos data]# nvidia-smi
Tue Aug 29 15:31:39 2017
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 384.66                 Driver Version: 384.66                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P40           On   | 00000000:00:03.0 Off |                    0 |
| N/A   22C    P8    10W / 250W |      0MiB / 22912MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla P40           On   | 00000000:00:06.0 Off |                    0 |
| N/A   23C    P8     9W / 250W |      0MiB / 22912MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

VNC Unavailable on Console

Last updated：2025-02-24 15:33:22

Phenomenon Description

When logging in to an instance via using VNC to log in to a Windows instance or using VNC to log in to a Linux instance, the login screen may fail to display any prompts, such as showing a **black screen** or **only the Windows logo**, as shown in the figure below:



Possible Reasons

1. A graphics driver is installed on the GPU instance.
   When you log in to a GPU instance via VNC, the default process accesses the QEMU-emulated VGA device to retrieve the operating system’s framebuffer. After the GPU graphics driver is installed, the framebuffer is no longer handled by the VGA device, making it inaccessible through VNC.

2. Due to other reasons, such as the installation of third-party software that conflicts with the operating system, the system fails to start successfully.

Solutions

1. If the instance has a public IP address, you can see Using RDP File to Log in to the Windows Instance for login instructions.
2. For GPU instances with a graphics driver installed, you can manually install a VNC Server on the instance. After installation, you can log in using a VNC Client from your local machine.
   Obtain the VNC Server/Client installation package independently.
3. If the issue is confirmed to be caused by reason 1, you can disable the GPU driver using a TAT command. After disabling the driver, you can log in to the instance through the console VNC. Navigate to Display Settings > Extend These Monitors > Show Only on 1. After completing the process, use the TAT command again to re-enable the GPU driver.
   For TAT commands, see Using TAT Command to Disable and Restore the Graphics Card Driver.
4. Check the installed third-party software to analyze potential reasons for the inability to log in to the instance via VNC.
   It is recommended to uninstall the third-party software or reinstall the system.