

Cloud GPU Service

Troubleshooting

Product Documentation



Copyright Notice

©2013–2026 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Troubleshooting

GPU Instance Troubleshooting Guide

Troubleshooting Common Xid Errors

Collecting Log for GPU Instances

GPU Usage Shows 100%

VNC Login Failures

Troubleshooting

GPU Instance Troubleshooting Guide

Last updated: 2026-01-19 12:02:36

This document provides guidance on handling GPU instance issues to help you quickly diagnose and resolve related problems. The following troubleshooting tips and recommendations cover common GPU instance issues.

System Status Check

For GPU servers, users are recommended to keep the GPU driver version up to date, disable the nouveau module, enable GPU driver persistence mode, and configure it to start automatically at system boot.

For GPU servers, the following configurations are recommended:

- Maintain an up-to-date and appropriate GPU driver version.
- Disable the nouveau module.
- Enable GPU Persistence Mode and configure it to start on boot.
- In the event of a GPU failure, we recommend restarting the instance via the Console to see if the issue can be 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 operating systems, it is recommended to directly select Linux 64-bit.
- Select a driver that is **recommended or certified** by NVIDIA.

Disabling the Nouveau Module

Nouveau is an open-source driver for NVIDIA graphics cards. It can conflict with the official NVIDIA GPU driver, so the nouveau module must be disabled on the system.

If the following command returns no output, the nouveau module has been successfully 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

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

Configuring GPU Persistence Mode

Enabling GPU Persistence Mode can mitigate various issues, such as GPU loss, bandwidth degradation, and temperature monitoring failures. We recommend enabling GPU Persistence Mode and configuring it to start on boot.

Common methods to check GPU Persistence Mode status:

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

Processes:						GPU Memory Usage
GPU ID	GI ID	CI ID	PID	Type	Process name	

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 1

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

Method 2

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

Configure autostart on startup:

```
# vim /etc/rc.d/rc.local
# Add the following line to the file:
# nvidia-smi -pm 1
# Grant executable permissions to the /etc/rc.d/rc.local file
# chmod +x /etc/rc.d/rc.local
# Reboot 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 Detected

To verify GPU detection status, first ensure that the `lspci` command detects all GPUs, and then ensure that the `nvidia-smi` command detects all GPUs.

Checking GPU Detection via `lspci`

Run the following command to ensure all GPUs are recognized correctly. Verify that each GPU entry ends with `(rev a1)`. If the output ends with `(rev ff)`, it indicates a GPU anomaly.

```
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 Detection via nvidia-smi

Run the following command to verify GPU detection:

```
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-MI  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 |   0B / 32510MiB |    0%    Default |
N/A  41C   P0   41W / 300W |   0MiB / 32510MiB |             |     N/A |
+-----+-----+-----+-----+-----+-----+
 1  Tesla V100-SXM2...  On  | 00000000:1B:00.0 Off |   0B / 32510MiB |    0%    Default |
N/A  36C   P0   42W / 300W |   0MiB / 32510MiB |             |     N/A |
+-----+-----+-----+-----+-----+-----+
 2  Tesla V100-SXM2...  On  | 00000000:3D:00.0 Off |   0B / 32510MiB |    0%    Default |
N/A  38C   P0   44W / 300W |   0MiB / 32510MiB |             |     N/A |
+-----+-----+-----+-----+-----+-----+
 3  Tesla V100-SXM2...  On  | 00000000:3E:00.0 Off |   0B / 32510MiB |    0%    Default |
N/A  37C   P0   42W / 300W |   0MiB / 32510MiB |             |     N/A |
+-----+-----+-----+-----+-----+-----+
 4  Tesla V100-SXM2...  On  | 00000000:88:00.0 Off |   0B / 32510MiB |    0%    Default |
N/A  37C   P0   41W / 300W |   0MiB / 32510MiB |             |     N/A |
+-----+-----+-----+-----+-----+-----+
 5  Tesla V100-SXM2...  On  | 00000000:89:00.0 Off |   0B / 32510MiB |    0%    Default |
N/A  38C   P0   40W / 300W |   0MiB / 32510MiB |             |     N/A |
+-----+-----+-----+-----+-----+-----+
 6  Tesla V100-SXM2...  On  | 00000000:B1:00.0 Off |   0B / 32510MiB |    0%    Default |
N/A  35C   P0   40W / 300W |   0MiB / 32510MiB |             |     N/A |
+-----+-----+-----+-----+-----+-----+

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

```

Note:

We recommend restarting the instance to attempt recovery. If the issue persists and the GPU status remains abnormal after the restart, 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 check the GPU bandwidth using the `lspci` or `nvidia-smi` command.

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 1

```
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 2

```
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 Standards](#).

Note:

You can contact the platform for a GPU replacement if the nvidia fielddiag test fails or if the GPU Retired Pages count meets any of the following conditions:

In the Retired Pages parameter:

- Double Bit ECC \geq 5 within 30 days.
- Double Bit ECC \geq 10 within the warranty period.
- Total Double Bit ECC + Single Bit ECC \geq 60 within the warranty period.

Methods to Query Retired Pages

Method 1

```
# 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 2

```
# 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
```

Recommendation

- If the GPU Retired Pages count meets NVIDIA RMA standards: Please contact the platform for hardware replacement.
- If the GPU Retired Pages count does not meet NVIDIA RMA standards: Check if the address space of the current error is masked (i.e., Pending: No). If not, restart the system or reset the GPU to mask the error

address, then retest the application. If the application is still affected by ECC errors after the address is masked, perform a fielddiag test. If the test fails, contact the platform for a GPU replacement.

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

Checking for GPU ERR! Status

During GPU operation, ERR! errors related to fans, power, etc., may occur. You can identify these by checking if the `nvidia-smi` output contains ERR!.

Example of Power ERR! Error:

```

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 ID                               Usage
+-----+-----+-----+-----+-----+-----+

```

Example of Fan ERR! Error:

```

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 identify this issue by checking if 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

Xid 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 driver programming for the GPU or corrupted commands sent to the GPU. For common Xid events and recommended solutions, see [Troubleshooting Common Xid Errors](#).

If the above status checks and common troubleshooting steps fail to resolve the issue, please contact the platform engineers for further troubleshooting assistance.

Troubleshooting Common Xid Errors

Last updated: 2026-01-19 12:03:00

This document explains what Xid messages are and provides explanations and troubleshooting methods for common Xid errors.

What Are Xid Messages?

Xid messages are error reports printed by the NVIDIA driver to the operating system's kernel log or event log. An Xid message indicates that a GPU error has occurred, typically caused by the driver programming the GPU incorrectly or corruption of commands sent to the GPU.

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 [online support](#).

Troubleshooting Common Xid Errors

Different Xid errors indicate different issues. Based on whether the user can resolve the issue independently, common Xid errors and their corresponding recommendations are classified into two categories below. For a complete list of Xid errors, please refer to the NVIDIA Xid Documentation.

Attempting Self-Resolution

When encountering the following Xid events, you can attempt to resolve them using the recommended solutions below. If the issue persists, you may provide feedback through [online support](#), where Tencent Cloud engineers are available 24/7 to assist you.

XID 48 Error

XID 48: Double Bit ECC Error

This error occurs when the GPU encounters an uncorrectable error. It is also reported to the user application. Typically, resetting the GPU or restarting the CVM instance is required to clear this error.

Recommendation: Restart the instance to recover. If the issue persists after a restart, please contact the platform for troubleshooting. **If your business is sensitive to Xid 48 errors, you can request a GPU replacement directly.**

XID 79 Error

XID 79: GPU has fallen off the bus

This error is generally caused by GPU driver or hardware issues. Users may observe that the GPU has detached from the instance (GPU loss).

Recommendation: Restart the instance to recover. If the issue persists after a restart, please contact the platform for troubleshooting.

XID 94 Error

XID 94: Contained ECC error

This indicates a contained ECC error on the GPU. Applications using the GPU will stop.

Recommendation: Restart the application to verify if the service returns to normal. If the application fails again, restart the instance. If the issue persists after a restart, please contact the platform for troubleshooting.

XID 95 Error

XID 95: Uncontained ECC error

This indicates an uncontained ECC error on the GPU. Applications using the GPU will stop.

Recommendation: Restart the instance to recover. If the issue persists after a restart, please contact the platform for troubleshooting.

XID 119 Error

XID 119: GSP RPC Timeout

This error is generally caused by the GPU driver triggering a bug in the GPU System Processor (GSP).

Recommendation:

1. Disable GSP. In newer generation instances, NVIDIA GPUs include GSP firmware functionality. GSP is designed to offload GPU initialization and other management tasks. You can follow the steps below to disable GSP: (For more details, please refer to [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:

- If you are using 535-series drivers, update to version 535.216.01 or later. If using 550-series drivers, update to version 550.144.03 or later. The newer driver versions address the XID 119 error issue 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.

Errors Requiring Platform Support

When encountering the following Xid errors, we recommend reporting them directly via [Online Support](#).

Tencent Cloud engineers are available 24/7 to assist you.

Recommendation: Refer to [Collecting Log for GPU Instances](#) to gather the necessary logs, and then contact the platform for troubleshooting.

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: 2026-01-19 12:03:22

This document provides guidance on collecting GPU instance logs to assist in analyzing and resolving GPU-related issues. The following instructions outline how to effectively collect these logs.

You can analyze the collected logs yourself or submit them 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.


```
[root@UM_18_107_centos data]# nvidia-smi -pm 1
Persistence mode is already Enabled for GPU 00000000:00:03.0.
Persistence mode is already Enabled for GPU 00000000:00:06.0.
All done.
[root@UM_18_107_centos data]# _
```

2. Run the following command to check GPU usage:

```
nvidia-smi
```

You will see the GPU usage is normal, as shown below:

```
[root@UM_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  |
+-----+-----+

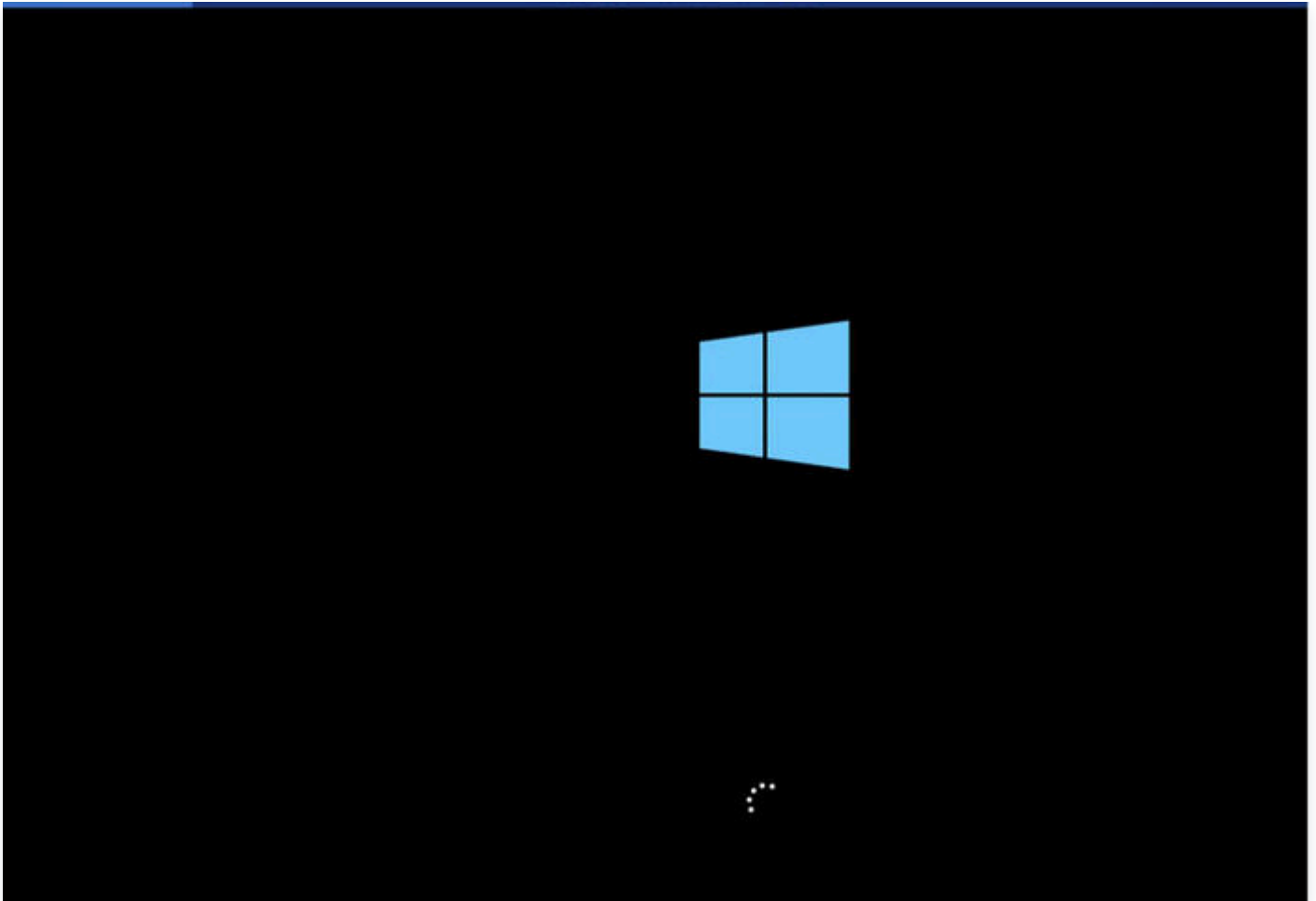
+-----+-----+
| Processes:                         GPU Memory          |
| GPU       PID  Type  Process name      Usage              |
+-----+-----+
| No running processes found         |
+-----+-----+
```

VNC Login Failures

Last updated: 2024-01-11 17:11:13

Problem Description

When you attempt to [log in to a Windows instance via VNC](#) or [log in to a Linux instance via VNC](#), you may not be prompted to log in, but instead encounter a **black screen** or the **blue Windows logo**, as shown below:



Possible Reasons

1. Your GPU instance is installed with a graphics driver.

When you log in to a GPU instance via VNC, the VGA device emulated by QEMU is accessed by default to obtain the framebuffer of the operating system for login. After you install a graphics driver on the GPU instance, the framebuffer will no longer be handled by the VGA device. As a result, you cannot log in to the operating system via VNC.

2. The operating system failed to start due to other causes. For example, third-party software that conflicts with the operating system is installed on the GPU instance.

Solution

1. If the GPU instance is installed with a graphics driver, install a VNC server on the instance so that you can log in to the GPU instance via a local VNC client.

You need to obtain the VNC server and the client installation packages by yourself.

2. Check the installed third-party software and analyze why the software leads to login failures.

We recommend that you uninstall the conflicting third-party software or reinstall the operating system.