GPU Cloud Computing
Operation Guide
Product Documentation
Copyright Notice

© 2013-2019 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 Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. 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

Operation Guide
  Best Practice
  Loging into Instance
  Restarting Instance
  Installing NVIDIA Driver
  Installing CUDA Driver
Security Group and Network

- Security group is a stateful virtual firewall for filtering packets. You can restrict access by using a firewall (security group) to allow the trusted addresses to access instances. Different security group rules for instance groups of different security levels are created to ensure that the instances running important business cannot be accessed easily from the outside.
- You need to regularly repair, update and protect the operating system and applications on the instance.
- With EIPs, you can quickly remap an address to another instance in your account (or NAT gateway instance) to block instance failures. For more information, see Elastic IP.
- Log in to user's Linux instances by use of SSH Key whenever possible. For the instances that you log in with password, the password needs to be changed from time to time.
- Choose to use Virtual Private Cloud to divide logical areas.

Storage

- For the data that requires high reliability, use Tencent Cloud's Cloud Block Storage to ensure the persistent and reliable data storage. Try not to select Local Disk for storage (Rendering GA2 only supports Cloud Block Storage).
- For databases that are frequently accessed and variable in size, use Tencent Cloud Database.
- You can use COS to store important data, such as static web pages, massive images and videos.

Backup and Recovery

- One of the recovery methods is to rollback a Custom Image you backed up via CVM Console.
- Deploy key components of an application across multiple availability zones, and copy the data as appropriate.
- Regularly view the monitoring data and set alarms as appropriate. For more information, see Cloud Monitor Product Documentation.
• Process emergent requests with Auto Scaling. For more information, see Auto Scaling Product Documentation.
Loging into Instance

Last updated: 2020-02-26 16:42:00

After you purchase and start a GPU instance, you can connect to and log in to it. The login method varies, depending on your local operating system, GPU instance operating system and whether the instance can be accessed via the Internet.

Preconditions

- To log in to a GPU instance using a password, you must have an admin account ID and password.
- To log in to a GPU instance using a key, you must create and download a private key.

Login Instruction

If your GPU instance is based on Linux, see Log in to Linux Instance

If your GPU instance is based on Windows, see Log in to Windows Instance
Restarting Instance

Last updated: 2019-12-01 15:56:31

Restart is usually used for the maintenance of GPU instances. Restarting a GPU instance is equivalent to restarting the operating system of a local PC.

Overview

- **Preparation for restart**: The instance cannot provide service during restart. Make sure GPU instance has stopped receiving service requests before restart.

- **How to perform restart**: It is recommended to restart an instance using the restart operation provided by Tencent Cloud, instead of running the restart command in the instance (such as restart command under Windows and Reboot command under Linux).

- **Restart period**: Generally, it takes only a few minutes to restart an instance.

- **Physical attributes of instances**: Restarting an instance does not change any of its physical attributes, so the public IP, private IP, and any data stored on the instance will remain unchanged.

- **Billing**: Restarting instances will not start a new billing period.

Restarting an Instance on the Console

1. Log in to the CVM Console.

2. Restart an instance: Select the instance to be restarted, and click **Restart** at the top of the list, or click **More -> CVM Status -> Restart** in the Operation column on the right side.

3. Restart multiple instances: Select all the instances to be restarted and click **Restart** at the top of the list. Reasons are given for instances that cannot be restarted.

Restart instance using API

For more information, see RebootInstances API.
The proper infrastructure software needs to be installed for GPU instance to work properly. For NVIDIA series GPU, there are two software packages to be installed:

1. Hardware driver that drives GPU to work.
2. Libraries required by upper-level applications.

If NVIDIA GPU is used for general computing, Tesla Driver and CUDA need to be installed. This document only describes how to install Tesla Driver.

For the convenience of users, users can select to pre-install a specific version of the driver and the image of CUDA in the image marketplace when creating the GPU instance.

Installing the Driver in Linux System

There are two ways to install the driver in Linux system:

1. Install the driver using Shell script. It works with all Linux distributions, including CentOS and Ubuntu.
2. Install the driver using packages. It works with various Linux distributions, such as the installation using DEB and RPM.

Regardless of the installation method, the Linux driver of NVIDIA Tesla GPU needs to compile the kernel module during the installation process, so the system is required to install gcc and the package which the compiling of Linux kernel module depends on, such as kernel-devel-$\$(uname -r)$.

**Installing the driver using Shell script**

2. Select an operating system and an installer package. Take P40 as an example, search for the driver and select the version of the driver to download.
Note:
If "Linux 64-bit" is selected for the "Operating System", the shell installation file will be downloaded. If a specific distribution is selected, the corresponding package installation file will be downloaded.
3. After jumping to the page of selected version, click **DOWNLOAD**.

4. After jumping again, if there is a page that requires personal information, you can skip it directly. When the following page appears, right click on **AGREE&DOWNLOAD** and copy the link address
5. Log in to the GPU instance, and paste the link address copied in the previous step to download the installer package by executing `wget` command. Or, you can download the NVIDIA installer package to a local system, and upload it to the GPU instance server.

```
[root@VM_0_4_centos ~]# wget http://us.download.nvidia.com/tesla/396.44/NVIDIA-Linux-x86_64-396.44.run
```

6. Add the execution permission to the installer package. For example, add the execution permission to the file named `NVIDIA-Linux-x86_64-396.44.run`:

```
chmod +x NVIDIA-Linux-x86_64-396.44.run
```
7. Install the corresponding gcc and kernel-devel packages for the current system.

```
sudo yum install -y gcc kernel-devel-xxx
```

`xxx` is the kernel version number, which can be viewed by using `uname -r`.

8. Follow the prompts after running the driver installer.

```
sudo /bin/bash ./NVIDIA-Linux-x86_64-396.44.run
```

9. Once the installation is finished, run `nvidia-smi`. The driver installation is successful if you see GPU information displayed (similar to what is shown below).

![NVIDIA-SMI Command Output](image)

**Installing the driver using DEB/RPM**

**Installing the driver using DEB**


2. Select the corresponding operating system that supports DEB, such as Ubuntu 16.04, and get the download link: `wget http://us.download.nvidia.com/tesla/396.44/nvidia-diag-driver-local-repo-ubuntu1604-396.44_1.0-1_amd64.deb`.

3. Run the command to install the package.

```
dpkg -i nvidia-diag-driver-local-repo-ubuntu1604-396.44_1.0-1_amd64.deb
```
4. Update the software package with the `apt-get` command.

   ```bash
   apt-get update
   ```

5. Run the `apt-get` command to install the driver.

   ```bash
   apt-get install cuda-drivers
   ```

6. Run the `reboot` command to restart.

7. If the correct information can be output when running `nvidia-smi`, the driver is installed successfully.

### Installing the driver using RPM


2. Select the corresponding operating system that supports RPM package, such as rhel 7.x. Get the download link: `wget http://us.download.nvidia.com/tesla/396.44/nvidia-diag-driver-local-repo-rhel7-396.44-1.0-1.x86_64.rpm`

3. Install the RPM package using the `rpm` command.

   ```bash
   rpm -i nvidia-diag-driver-local-repo-rhel7-396.44-1.0-1.x86_64.rpm
   ```

4. Clear the cache using the `yum` command.

   ```bash
   yum clean all
   ```

5. Install the driver using the `yum` command.

   ```bash
   yum install cuda-drivers
   ```

6. Run the `reboot` command to restart.

7. If the correct information can be output when running `nvidia-smi`, the driver is installed successfully.

### Installing the Driver in Windows System


2. Select an operating system and an installer package. Suppose we use M40, we select the following driver:
3. Open the folder where the downloaded driver is located and double click the installation file to launch the installation program. Install the driver according to instructions and restart your instance when required. Check the Device Manager to verify whether the GPU is functioning normally.

**Reason for installation failure**

The failure of Linux system driver installation is reflected in the fact that nvidia-smi does not work. There are several common reasons for failure:

1. The system lacks the packages needed to compile the kernel module, such as gcc, kernel-devel-xxx, etc., resulting in failure to compile and install.
2. There are multiple versions of the kernel in the system. Due to the incorrect configuration of DKMS, the driver is compiled into a kernel module that is not the current version of the kernel, resulting in the failure of kernel module installation.
3. After the driver is installed, the original installation becomes invalid due to the upgrade of the kernel version.
Installing CUDA Driver

Last updated: 2019-08-06 18:41:27

CUDA (Compute Unified Device Architecture) is a computing platform published by the graphic card vendor NVIDIA. As a generic parallel computing architecture, CUDA(tm) allows GPUs to solve complex computing problems. It contains CUDA instruction set architecture (ISA) and parallel computing engine within the GPU. Now, developers can write programs for CUDA(tm) architecture using C, C++, and FORTRAN. These programs can be run with great performance on processors that support CUDA(tm).

GPU instances use NVIDIA graphic cards and you need to install CUDA development operating environment for them. Here we take the most commonly used CUDA 7.5 as an example. You can install it by following these steps.

In Linux System


2. Select an operating system and an installer package. Suppose we use CentOS 7.2 (64 bit), we choose as follows:

![Select Target Platform](image)
Notes:
It is recommended to choose rpm (network) for "Installer Type".
network: Network installer package. The package is small, and you need to download the actual installer package in the CVM via private network.
local: local installer package. The package is large because it contains the installer packages for all downloaded installation components.

3. Right click **Download** -> **Copy Link Address**.

![Download Target Installer for Linux CentOS 7 x86_64](image)

4. Log in to the GPU instance, and paste the link address copied in the previous step to download the installer package by executing `wget` command. Or, you can download the CUDA installer package to a local system, and upload it to the GPU instance server.

![wget command output](image)

5. Run the following commands in the directory where the CUDA installer package is located:

```
sudo rpm -i cuda-repo-rhel7-7.5-18.x86_64.rpm
```
6. Go to the `/usr/local/cuda-7.5/samples/1Utilities/deviceQuery` directory and execute `make` command to compile the deviceQuery program.

7. Execute deviceQuery. The CUDA installation is successful if the following device information is displayed.

!!deviceQuery Starting...

CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: “Tesla M40 24GB”
CUDA Driver Version / Runtime Version  8.0 / 7.5
CUDA Capability Major-Minor version number:  5.2
Total amount of global memory:  24605 MBytes (25695692736 bytes)
(C24) Multiprocessors, (128) CUDA Cores-MP:  3072 CUDA Cores
GPU Max Clock rate:  1112 Mhz (1.11 Gts)
Memory Clock rate:  3004 Mhz
Memory Bus Width:  384-bit
L2 Cache Size:  3145728 bytes
Maximum Texture Dimension Size (x,y,z):  ID=(65536), 2D=(65536, 65536), 3D=(4096, 4096, 4096)
Maximum Layered 2D Texture Size, (num) layers:  ID=(16384), 2D=(16384, 16384), 2048 layers
Total amount of constant memory:  65536 bytes
Total amount of shared memory per block:  49152 bytes
Total number of registers available per block:  46336
Warp size:  32
Maximum number of threads per multiprocessor:  2048
Maximum number of threads per block:  1024
Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
Max dimension size of a grid size (x,y,z): (2147483647, 65535, 65535)
Maximum memory pitch:  2147483647 bytes
Texture alignment:  512 bytes
Concurrent copy and kernel execution:  Yes with 2 copy engine(s)
Run time limit on kernels:  No
Integrated GPU sharing Host Memory:  No
Support host page-locked memory mapping:  Yes
Alignment requirement for Surfaces:  Yes
Device has ECC support:  Disabled
Device supports Unified Addressing (UVA):  Yes
Device PCI Domain ID / Bus ID / Location ID:  0 / 0 / 7

Compute Mode:  C (Default (multiple host threads can use ::cudaSetDevice() with device simultaneously))

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 8.0, CUDA Runtime Version = 7.5, NumDevs = 1, Device0 = Tesla M40
Result = PASS

In Windows System

To install CUDA on Windows instance, use the remote desktop to log in to your Windows instance as admin.

1. Download the CUDA installer package from CUDA Driver Official Website.
2. Select an operating system and an installer package. Suppose we use Win Server 2012 R2 (64 bit), we choose as follows:
3. Launch the installation program and proceed according to the instructions. Installation is successful when you see the final dialog indicating the end of the process.