GPU Cloud Computing
Operation Guide
Product Documentation
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: 2021-03-25 16:52:10

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](#).
Installing NVIDIA Driver

Last updated: 2020-09-22 16:39:14

Overview

The GPU instance must be installed with the necessary infrastructure software in advance. For an NVIDIA GPU instance, the following software packages are required:

- Hardware driver for the GPU
- Libraries required by upper-level applications

To use NVIDIA GPU instances for general computing tasks, you must install Tesla driver and Compute Unified Device Architecture (CUDA) driver. This document only describes how to install a Tesla driver. For more information on CUDA driver, please see Installing CUDA Driver.

Directions

Installing an NVIDIA Tesla driver on a Linux instance

You can use the Shell script to install a driver on the Linux instance. This method is applicable to all Linux distributions, including CentOS and Ubuntu.

When installing an NVIDIA Tesla driver for Linux, the driver needs to compile the kernel module. You must install gcc and packages required to compile the Linux kernel module in advance, such as `kernel-devel-$\text{(uname -r)}$`.

1. Run the following command to check whether dkms has been installed in the operating system:

   ```bash
   rpm -qa | grep -i dkms
   ```

   If the returned result is as shown in the following figure, dkms has been installed.

   ![dkms installed](image)

   If dkms is not installed, run the following command to install dkms:

   ```bash
   sudo yum install -y dkms
   ```

3. Configure the GPU type and operating system, and click **SEARCH** to search for the driver you need to download, as shown in the following figure. Below uses Tesla V100 as an example.

**Note:**
You can configure **Operating System** as **Linux 64-bit** to download shell setup files. If you configure **Operating System** to a specific Linux distribution, the corresponding installation files will be downloaded.

4. Select the required version to go to the driver download page, and click **DOWNLOAD**, as shown in the following figure.
5. You can skip the page for entering personal information. If the following page appears, right-click **AGREE & DOWNLOAD** and select **Copy link address**.

6. To log in to GPU instances, see Log into Linux Instance Using Standard Login Method. You can also use other login methods:
   - Logging into Linux instances via remote login tools
   - Logging into Linux instance via SSH key
7. Run the `wget` command to download the installation package using the URL copied in Step 5, as shown in the following figure.

```bash
[root@VM_0_116_centos ~]# wget http://us.download.nvidia.com/tesla/418.126.02/NVIDIA-Linux-x86_64-418.126.02.run
```

You can also download the installation package to your local computer and upload it to the GPU instance.
8. Add execution permissions to the installation package. For example, run the following command to add execution permissions to the `NVIDIA-Linux-x86_64-418.126.02.run` file:

```
chmod +x NVIDIA-Linux-x86_64-418.126.02.run
```

9. Run the following commands in sequence to check whether kernel-devel and gcc have been installed in the operating system:

```
rpm -qa | grep kernel-devel
rpm -qa | grep gcc
```

If the returned result is as shown in the following figure, kernel-devel and gcc have been installed.

```
[root@VM_0_116_centos ~]# rpm -qa | grep kernel-devel
kernel-devel-3.10.0-1062.9.1.el7.x86_64
[root@VM_0_116_centos ~]# rpm -qa | grep gcc
gcc-4.8.5-39.el7.x86_64
libgcc-4.8.5-39.el7.x86_64
```

If kernel-devel and gcc are not installed, run the following command to install them:

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

⚠️ **Note:**

If the kernel version has been upgraded, you must upgrade kernel-devel to the same version.

0. Run the following command to install the driver as instructed:

```
sudo sh NVIDIA-Linux-x86_64-418.126.02.run
```

1. After the installation is completed, run the following command to verify.

```
nvidia-smi
```

If GPU information similar to that shown in the following figure is returned, the installation is successful.
Installing an NVIDIA Tesla driver on a Windows instance

1. To log in to GPU instances, see Logging in to a Windows Instance Using the RDP File (Recommended).
2. Go to NVIDIA Driver Downloads.
3. Configure the GPU type and operating system, and click SEARCH to search for the driver you need to download, as shown in the following figure. Below uses Tesla V100 as an example.

   ![NVIDIA Driver Downloads](image)

4. Go to the directory where the downloaded installation package is located, double-click on it to install the driver as instructed, and restart the GPU instance as required.
After the installation is completed, go to **Device Manager** to check whether the GPU works properly.

**Reasons for installation failures**

If `nvidia-smi` does not run properly, the driver has not been installed correctly. Common reasons include:

1. The operating system does not have the required packages installed for compiling the kernel module, such as `gcc` and `kernel-devel`.
2. The operating system has kernels in multiple versions. Due to incorrect DKMS configuration, the driver compiles a kernel module that is not in the version of the current kernel, causing kernel module installation to fail.
3. After the driver is installed, kernel version upgrade causes the original installation to fail.
Installing CUDA Driver

Overview

Compute Unified Device Architecture (CUDA™) is a computing platform developed by NVIDIA. With a generic parallel computing architecture, CUDA allows GPUs to solve complex computing problems. It includes the CUDA instruction set architecture (ISA) and the parallel computing engine within the GPU. The CUDA platform is designed to work with programming languages such as C, C++, and Fortran. The compiled programs can be run on CUDA-enabled processors.

Because GPU instances use NVIDIA graphic cards, you must install the CUDA Toolkit. This document uses the most common CUDA Toolkit 10.1 as an example to describe how to install CUDA Toolkit on a GPU instance.

Directions

Installing CUDA Toolkit on a Linux instance

2. Select the CUDA Toolkit version, as shown in the following figure. Below uses CUDA Toolkit 10.1 as an example.

![CUDA Toolkit Archive](https://developer.nvidia.com/cuda-toolkit-archive)

3. Configure the platform information as instructed, as shown in the following figure.
4. When the following information appears, right-click **Download** and select **Copy link address**, as shown in the following figure.

5. To log in to GPU instances, see **Log into Linux Instance Using Standard Login Method**. You can also use other login methods:
   - **Logging into Linux instances via remote login tools**
   - **Logging into Linux instance via SSH key**

6. Run the **wget** command to download the installer using the URL copied in **Step 4**, as shown in the following figure.
You can also download the installer to your local computer and upload it to the GPU instance.

7. Add execution permissions to the installer. For example, to add execution permissions to the `cuda_10.1.105_418.39_linux.run` file, run the following commands in sequence:

```bash
sudo chmod +x cuda_10.1.105_418.39_linux.run
./cuda_10.1.105_418.39_linux.run --toolkit --samples --silent
```

8. Restart the operating system.

9. Run the following commands in sequence to configure environment variables:

```bash
echo 'export PATH=/usr/local/cuda/bin:$PATH' | sudo tee /etc/profile.d/cuda.sh
source /etc/profile
```

0. Run the following commands in sequence to check whether CUDA Toolkit has been installed:

```bash
cd /usr/local/cuda-10.1/samples/1Utilities/deviceQuery
make
./deviceQuery
```

If Result=PASS is returned, the installation is successful.

After you run the `make` command, the following error is shown.

```
[root@VM-32-212-centos deviceQuery]# make
/usr/local/cuda/bin/nvcc -ccbin g++ -I../../../../common/inc -m64 -gencode arch=compute_30,code=sm_30 -gencode arch=compute_35,code=sm_35 -gencode arch=compute_37,code=sm_37 -gencode arch=compute_50,code=sm_50 -gencode arch=compute_52,code=sm_52 -gencode arch=compute_60,code=sm_60 -gencode arch=compute_61,code=sm_61 -gencode arch=compute_70,code=sm_70 -gencode arch=compute_75,code=sm_75 -gencode arch=compute_75,code=compute_75 -o deviceQuery.o -c deviceQuery.cpp
g++: No such file or directory
make: *** [deviceQuery.o] Error 1
```

In this case, run the following command to install gcc:

```
yum install -y gcc-c++
```

After the installation is completed, repeat Step 10 to verify.
Installing CUDA Toolkit on a Windows instance

1. To log in to GPU instances, see Logging in to a Windows Instance Using the RDP File (Recommended).

2. Visit the CUDA Toolkit download page.

3. Select the CUDA Toolkit version, as shown in the following figure. Below uses CUDA Toolkit 10.1 as an example.

4. Configure the platform information as instructed, as shown in the following figure.

5. Go to the directory where the downloaded installer is located, double-click on it to install CUDA Toolkit as instructed, and restart the GPU instance as required.
If the dialog box shown in the following figure appears, CUDA Toolkit has been installed.