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
  Logging In to Instances
  Restarting Instances
  Installing NVIDIA Driver
  Installing CUDA Driver
Operation Guide

Best Practice

Last updated : 2019-09-19 17:13:40

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.
Logging In to Instances

After you purchase and launch 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

For Linux-based GPU instances, see Logging In to Linux Instances

For Windows-based GPU instances, see Logging In to Windows Instances
Restarting an instance is a common approach for troubleshooting. Restarting a GPU instance is equivalent to restarting the operating system of a local PC.

Overview

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

- **How to restart:** It is recommended to restart an instance in the Tencent Cloud console, 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 in the Console

1. Log in to the CVM Console.

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

3. To restart multiple instances, select all the instances to restart, and click Restart at the top of the list.

Restarting an Instance Using API

For more information, see RebootInstances API.
Installing NVIDIA Driver

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-$(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.

   ```bash
   [root@VM_0_116_centos ~]# rpm -qa | grep -i dkms
   dkms-2.8.1-4.20200214git5ca628c.el7.noarch
   ```

   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.

![NVIDIA Driver Downloads](image)

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.

![GPU Information]

### 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]

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.
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](image)

   Previous releases of the CUDA Toolkit, GPU Computing SDK, documentation and developer drivers can be found using the links below. Please select the release you want from the list below, and be sure to check www.nvidia.com/drivers for more recent production drivers appropriate for your hardware configuration.

   **Latest Release**
   - CUDA Toolkit 11.0 [May 2020], Versioned Online Documentation
   - CUDA Toolkit 10.2 [Nov 2019], Versioned Online Documentation
   - CUDA Toolkit 10.1 update2 [Aug 2019], Versioned Online Documentation
   - CUDA Toolkit 10.1 update1 [May 2019], Versioned Online Documentation
   - CUDA Toolkit 10.1 [Feb 2019], Online Documentation
   - CUDA Toolkit 10.0 [Sept 2018], Online Documentation

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.

**Note:**

**Installer Type:** We recommend selecting **runfile (local)**.

- **network:** the network installer. It is a small executable and an internet connection is required during the installation.
- **local:** the local installer. It is very large and has all of the components embedded into it.
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:

```
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:

```
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:

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

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

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

```
[...]
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.

![CUDA Toolkit Archive](image)

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

![Select Target Platform](image)

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.