Cloud GPU Service

Operation Guide

Product Documentation

Contents

Operation Guide
Best Practice

Last updated：2024-01-11 17:11:13

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

Last updated：2024-01-11 17:11:13

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 Instances

Last updated：2024-01-11 17:11:13

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

Installing NVIDIA Driver

Last updated：2024-03-27 11:47:40

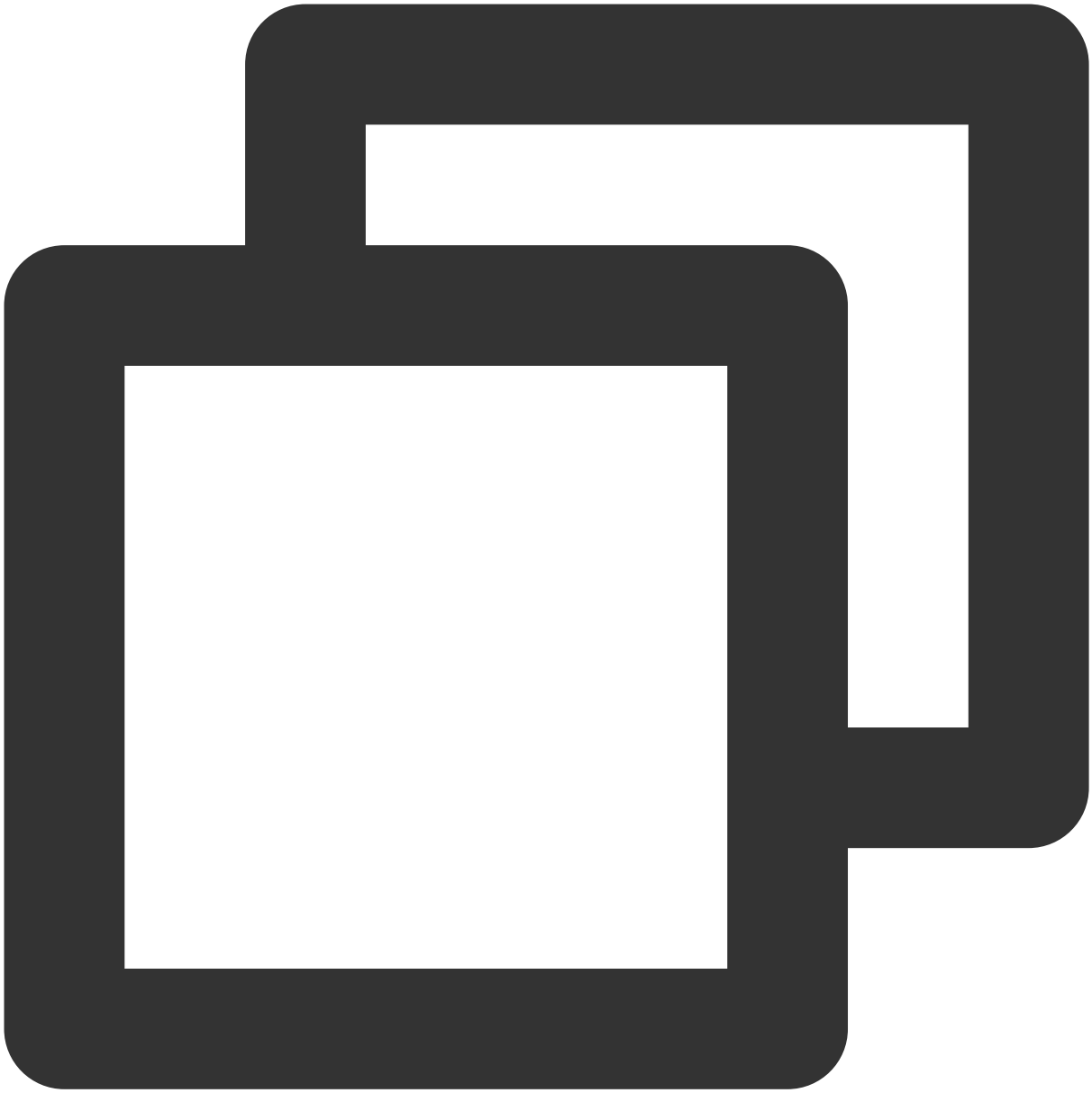
Overview

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

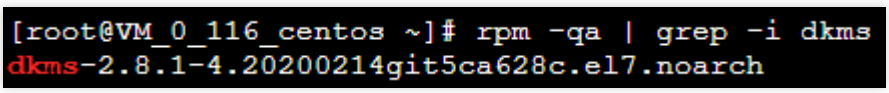
Installing an NVIDIA Tesla driver on a Linux instance

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:

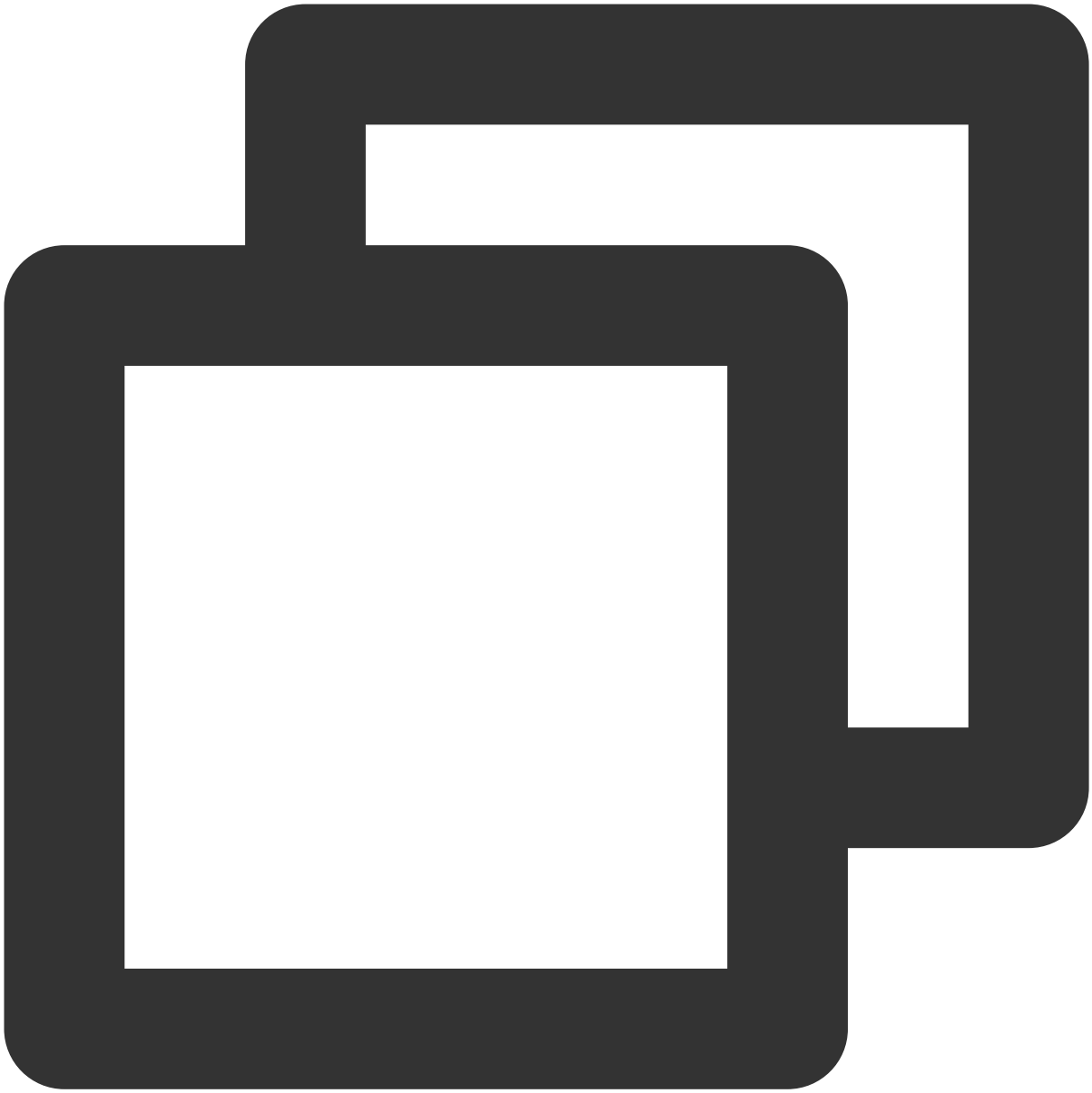


```
rpm -qa | grep -i dkms
```

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



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



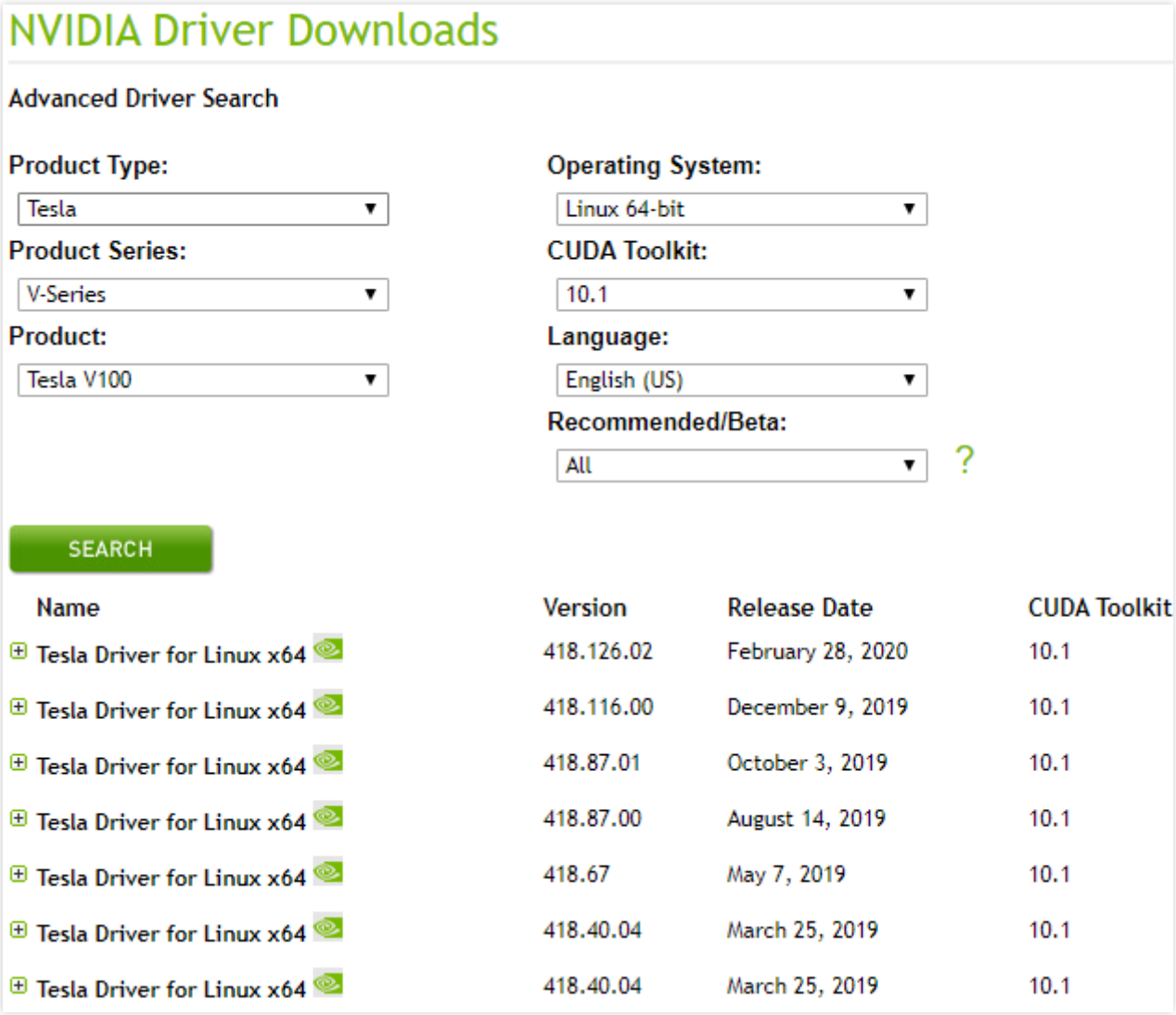
```
sudo yum install -y dkms
```

2. Go to NVIDIA Driver Downloads or visit `http://www.nvidia.com/Download/Find.aspx` .

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.

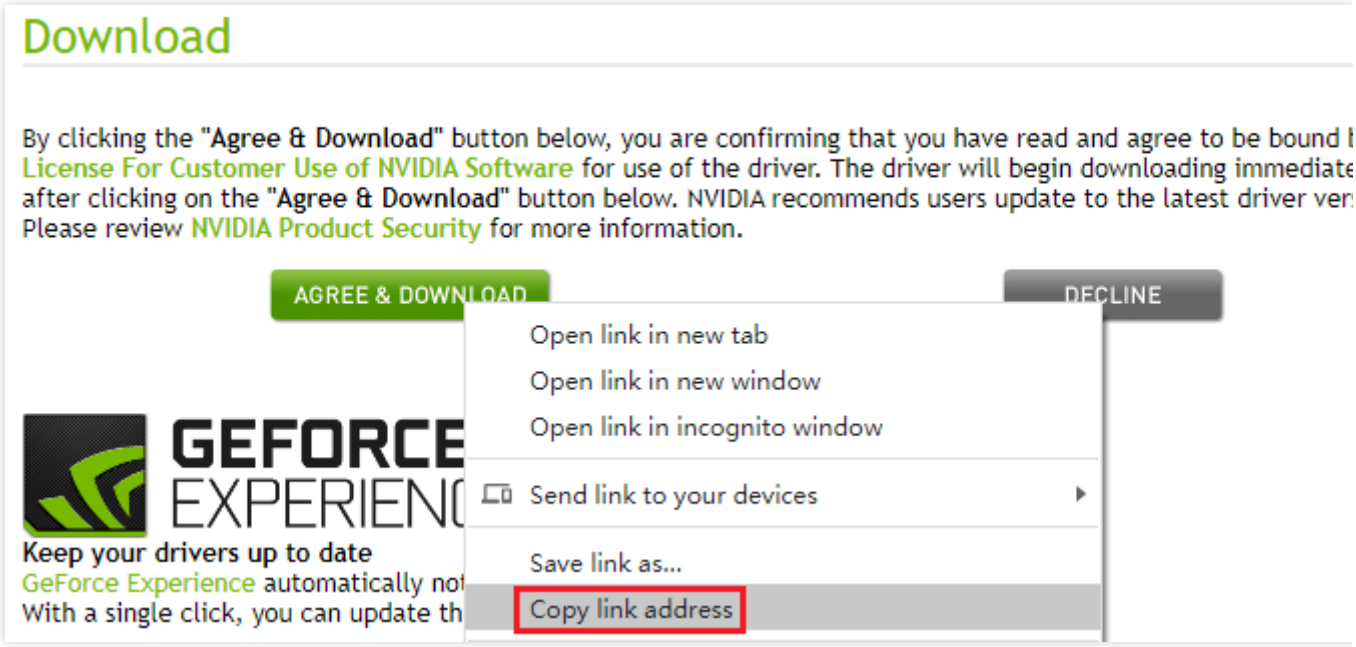


4. Select the required version to go to the driver download page, and click **DOWNLOAD**, as shown in the following figure.

5.

You

can skip the page for entering personal information. If the following page appears, right-click **AGREE & DOWNLOAD** and select **Copy link address**.

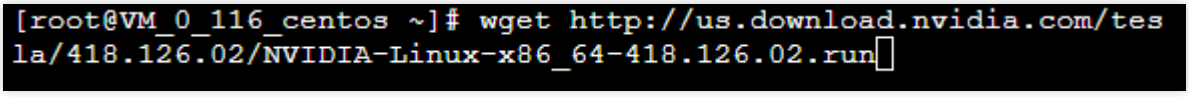


6. To log in to GPU instances, see Log into Linux Instance Using Standard Login Method. You can also use other login methods:

Logging into Linux instances via remote login tools

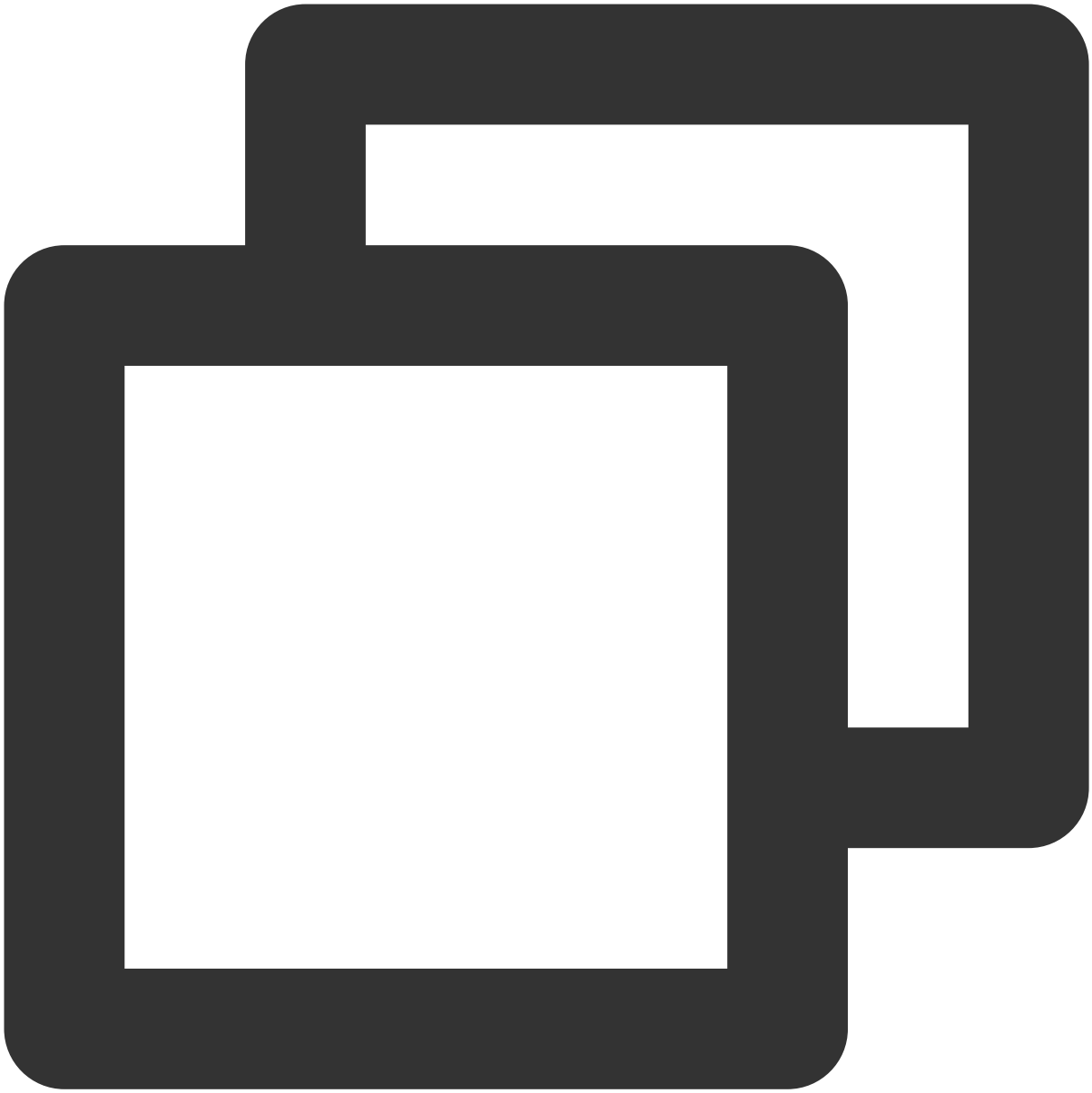
Logging into Linux instance via SSH key

7. Run the `wget` command to download the installation package using the URL copied in Step 5, as shown in the following figure.



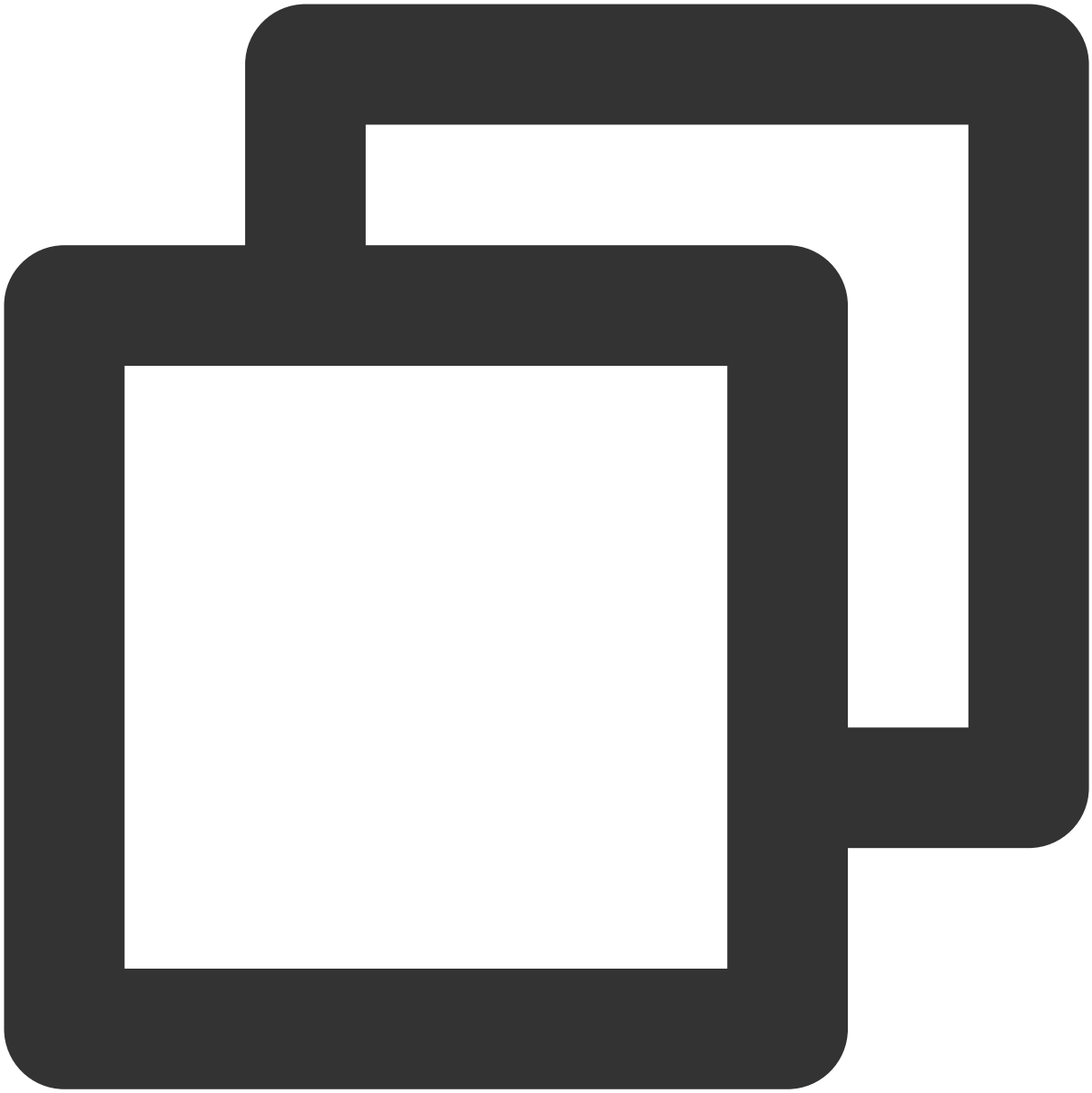
You can also download the installation package to your local computer and upload it to the GPU instance.

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:

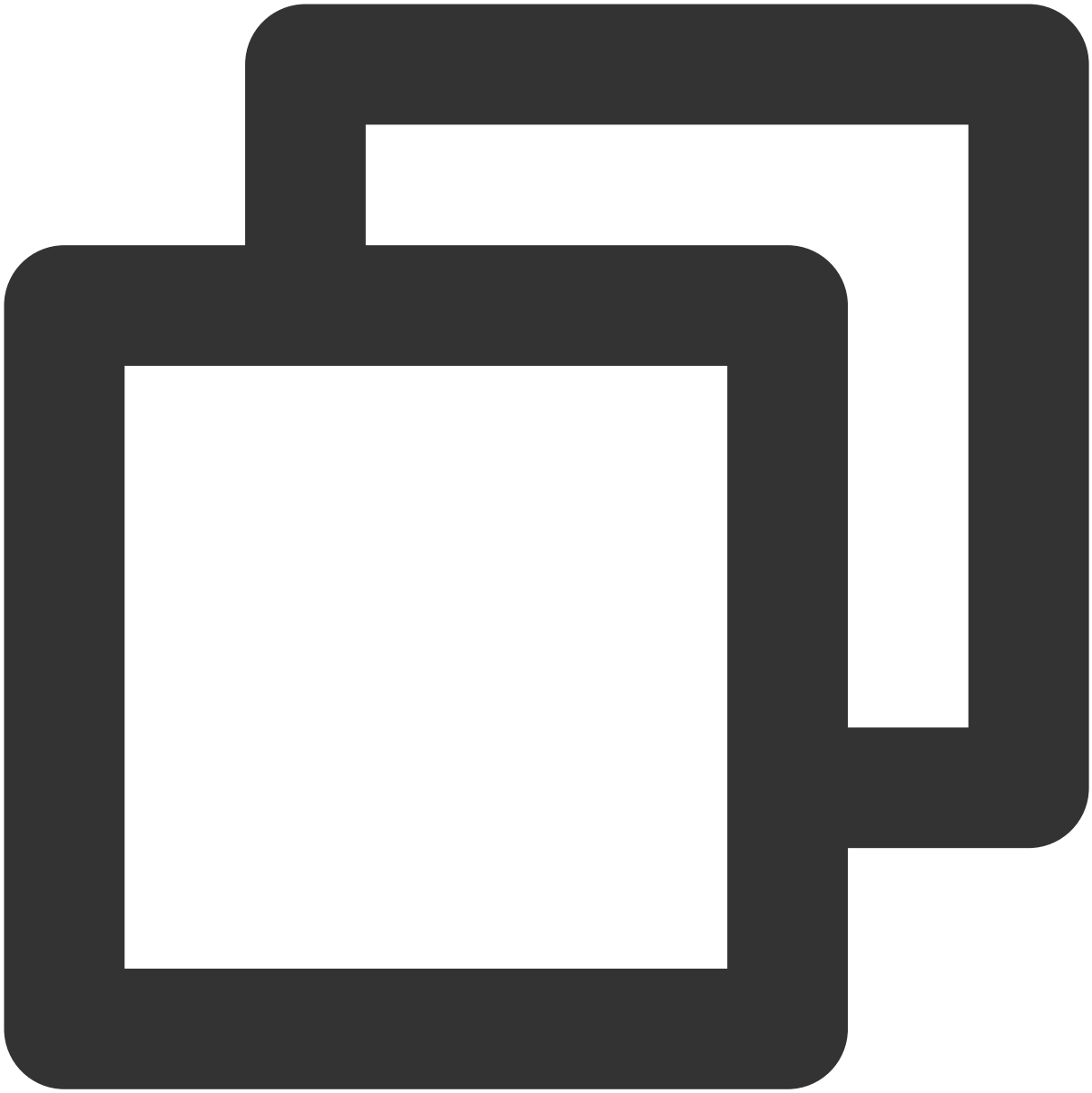


```
chmod +x NVIDIA-Linux-x86_64-418.126.02.run
```

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
```

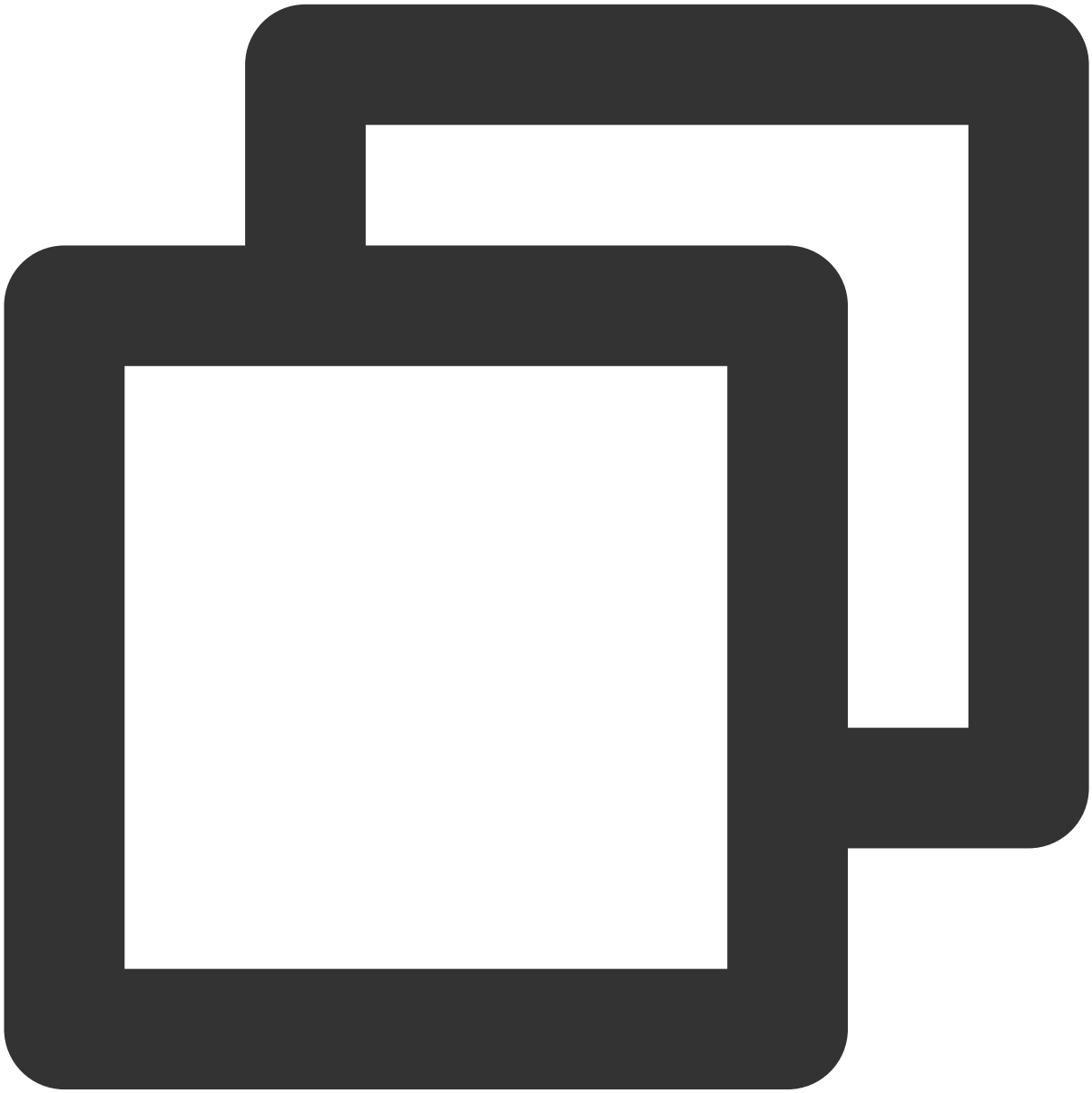


```
rpm -qa | grep gcc
```

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:

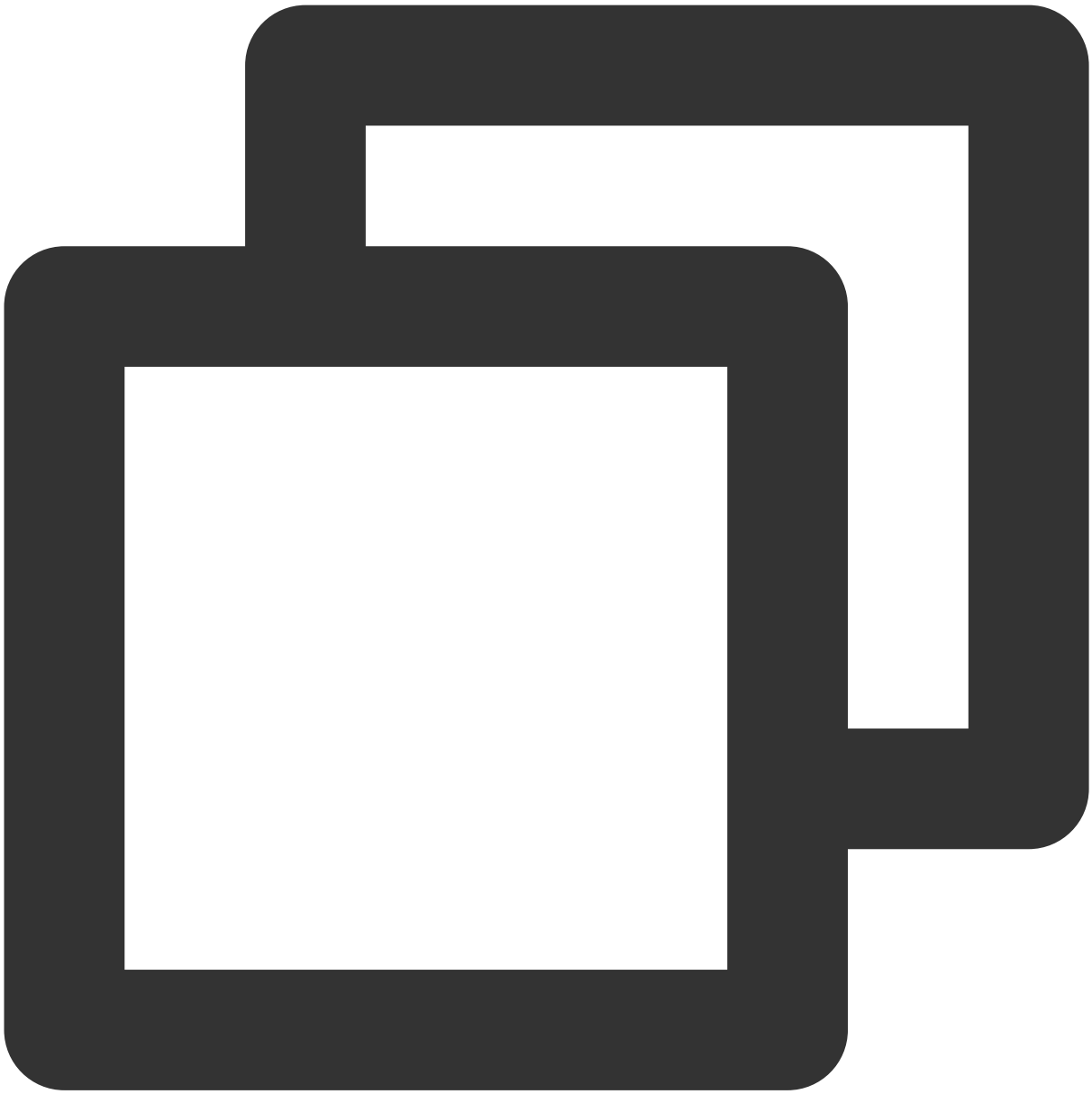


```
sudo yum install -y gcc kernel-devel
```

Note:

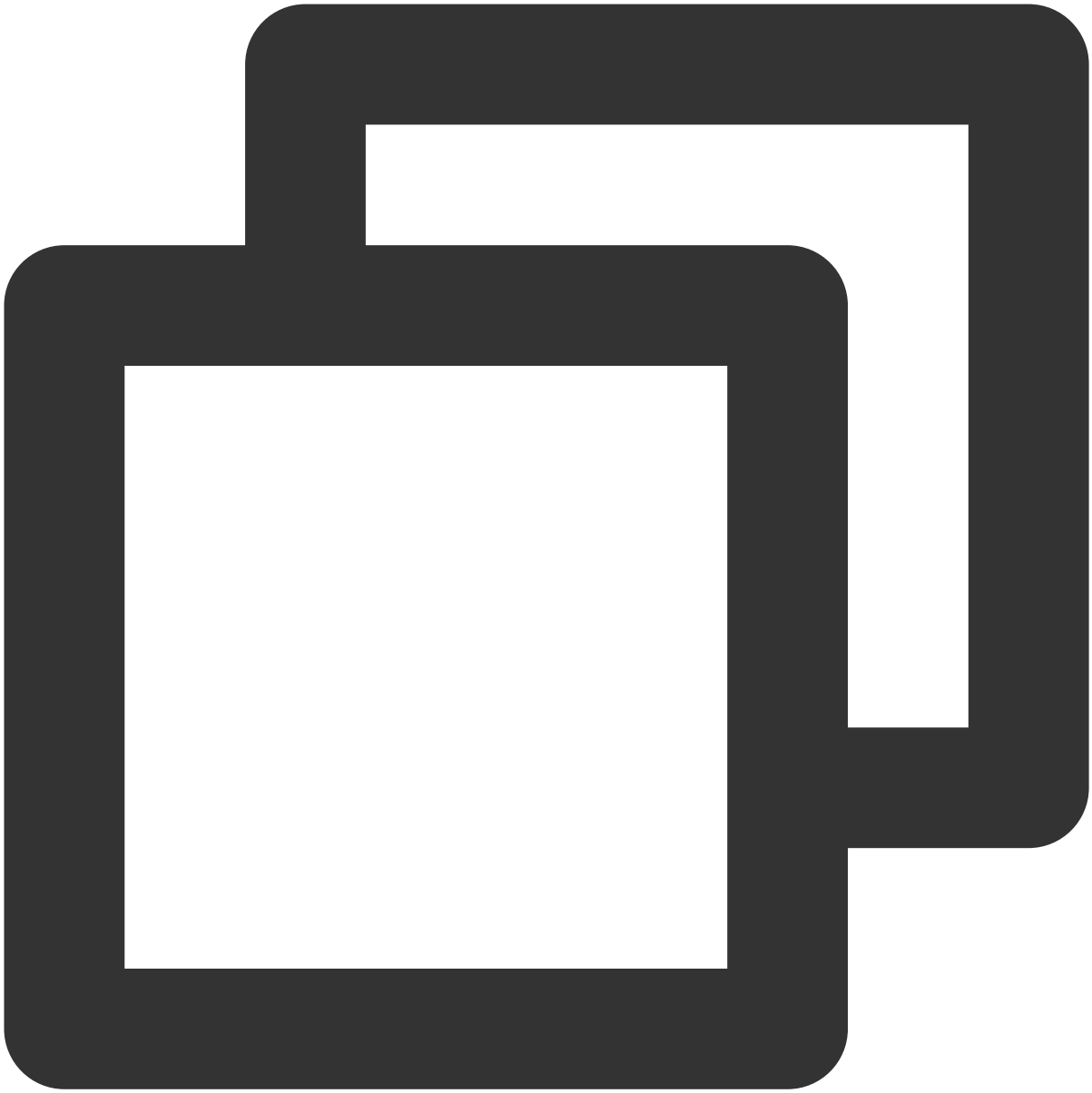
If the kernel version has been upgraded, you must upgrade kernel-devel to the same version.

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



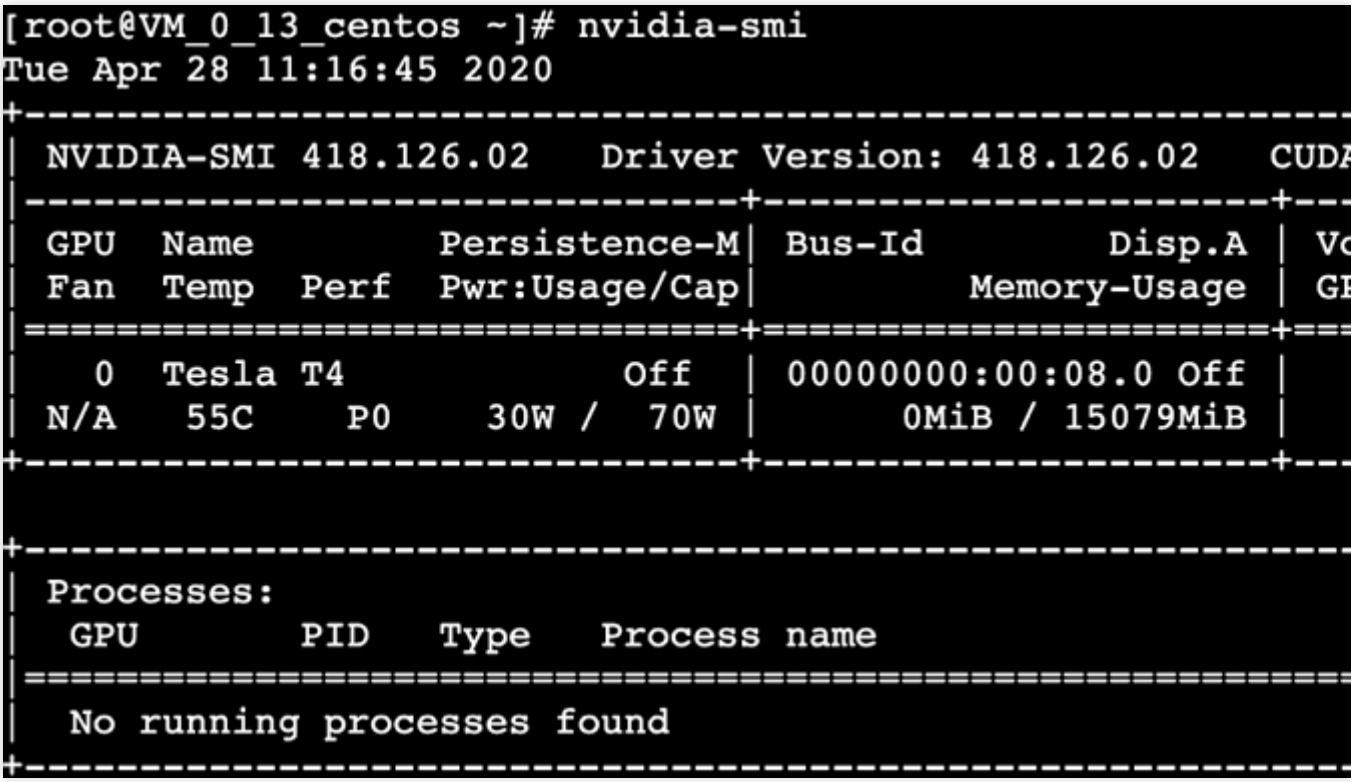
```
sudo sh NVIDIA-Linux-x86_64-418.126.02.run
```

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



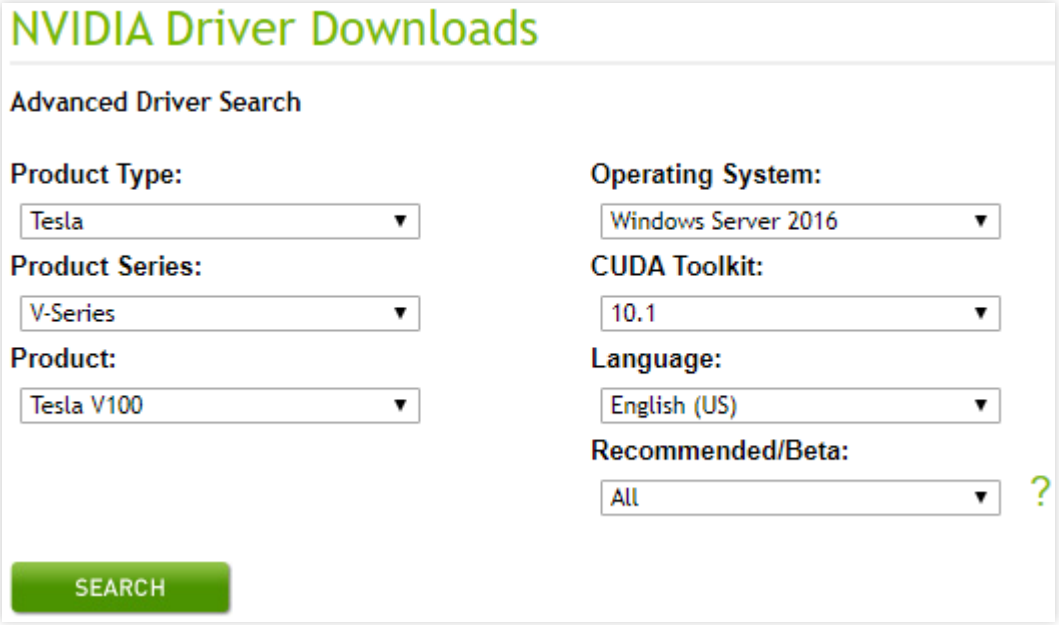
```
nvidia-smi
```

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.



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

Last updated：2024-01-11 17:11:13

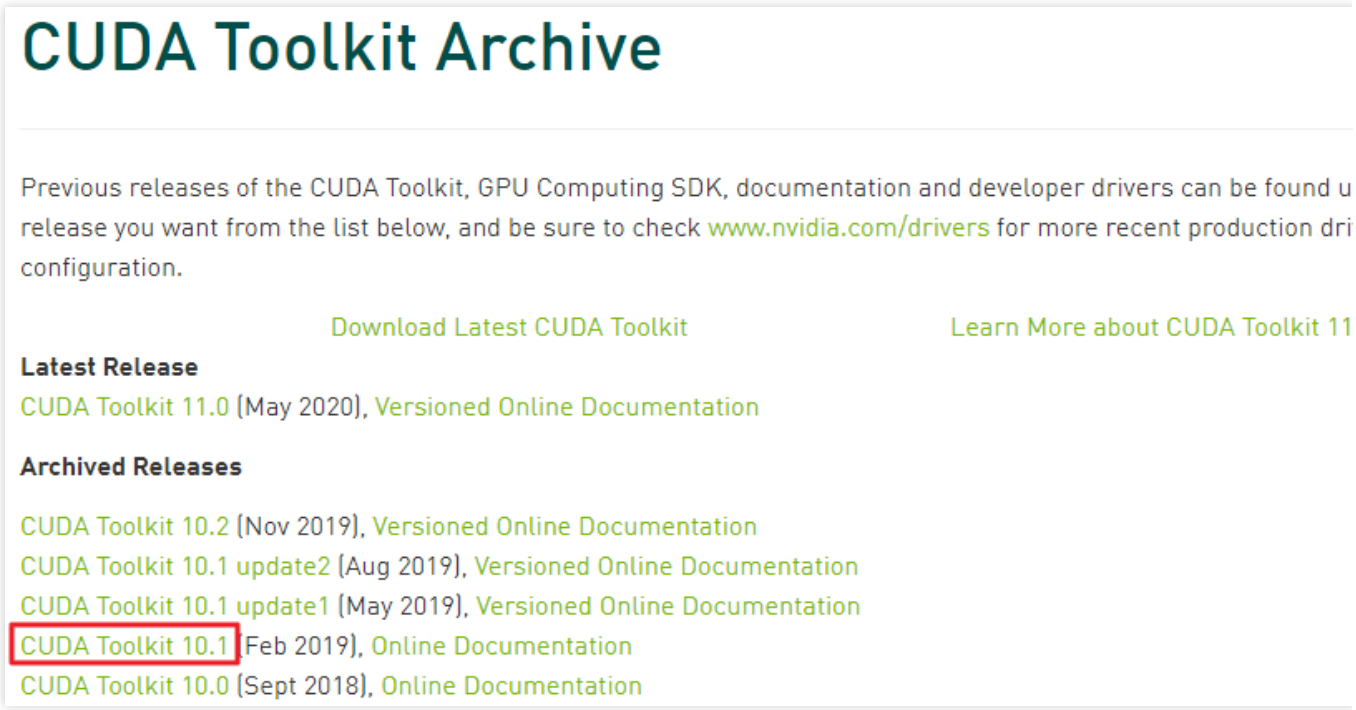
Overview

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

1. Go to the CUDA Toolkit download page or visit `https://developer.nvidia.com/cuda-toolkit-archive` .
2. Select the CUDA Toolkit version, as shown in the following figure. Below uses CUDA Toolkit 10.1 as an example.



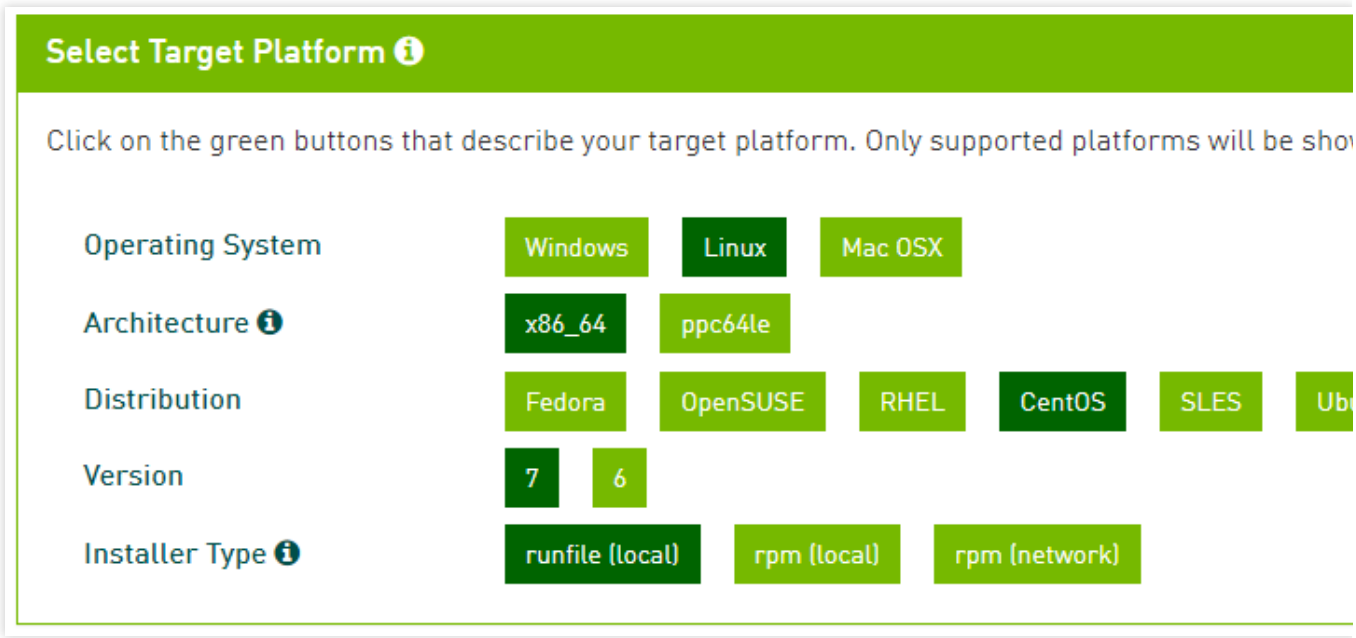
3. Configure the platform information as instructed, as shown in the following figure.

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.

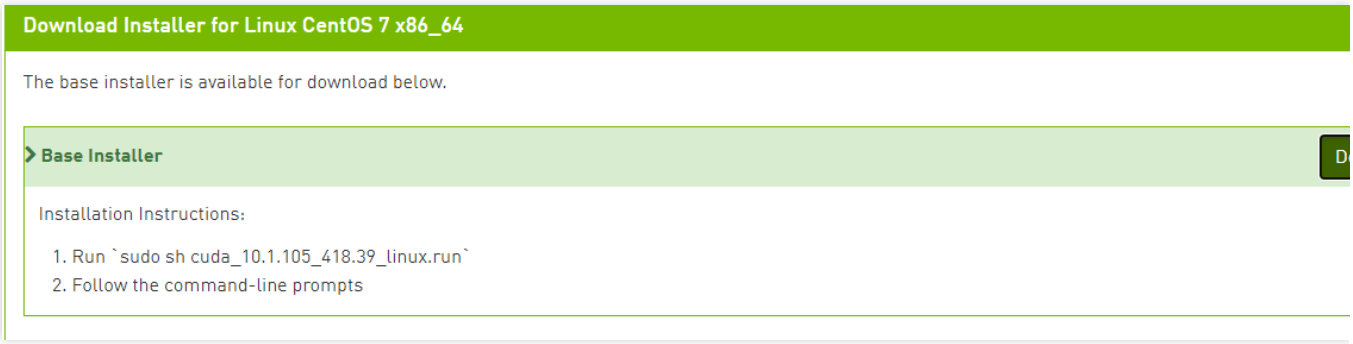
local: the local installer. It is very large and has all of the components embedded into it.



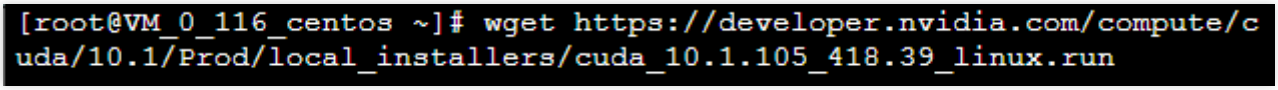
4.

Whe

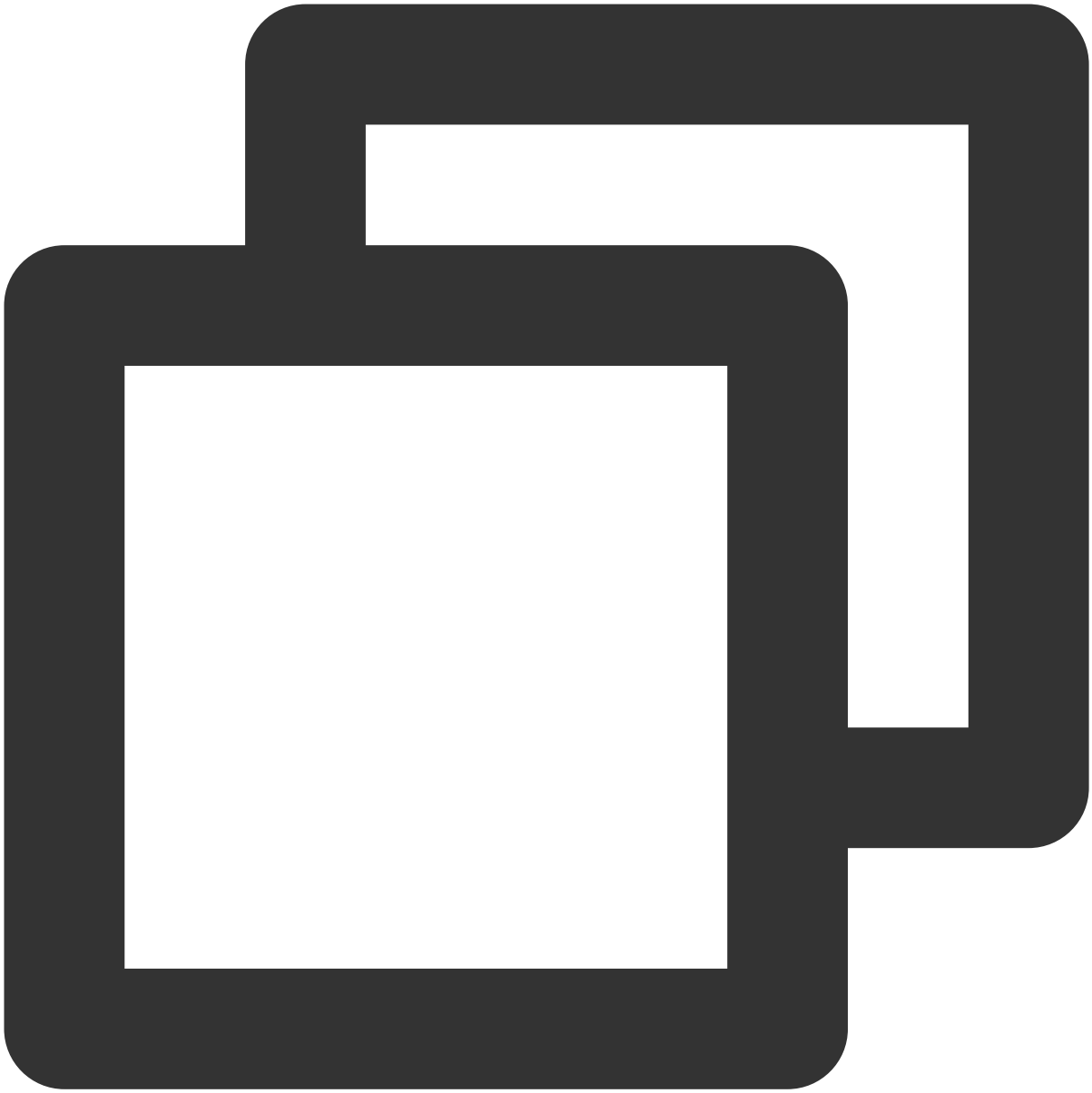
n the following information appears, right-click **Download** and select **Copy link address**, as shown in the following figure.



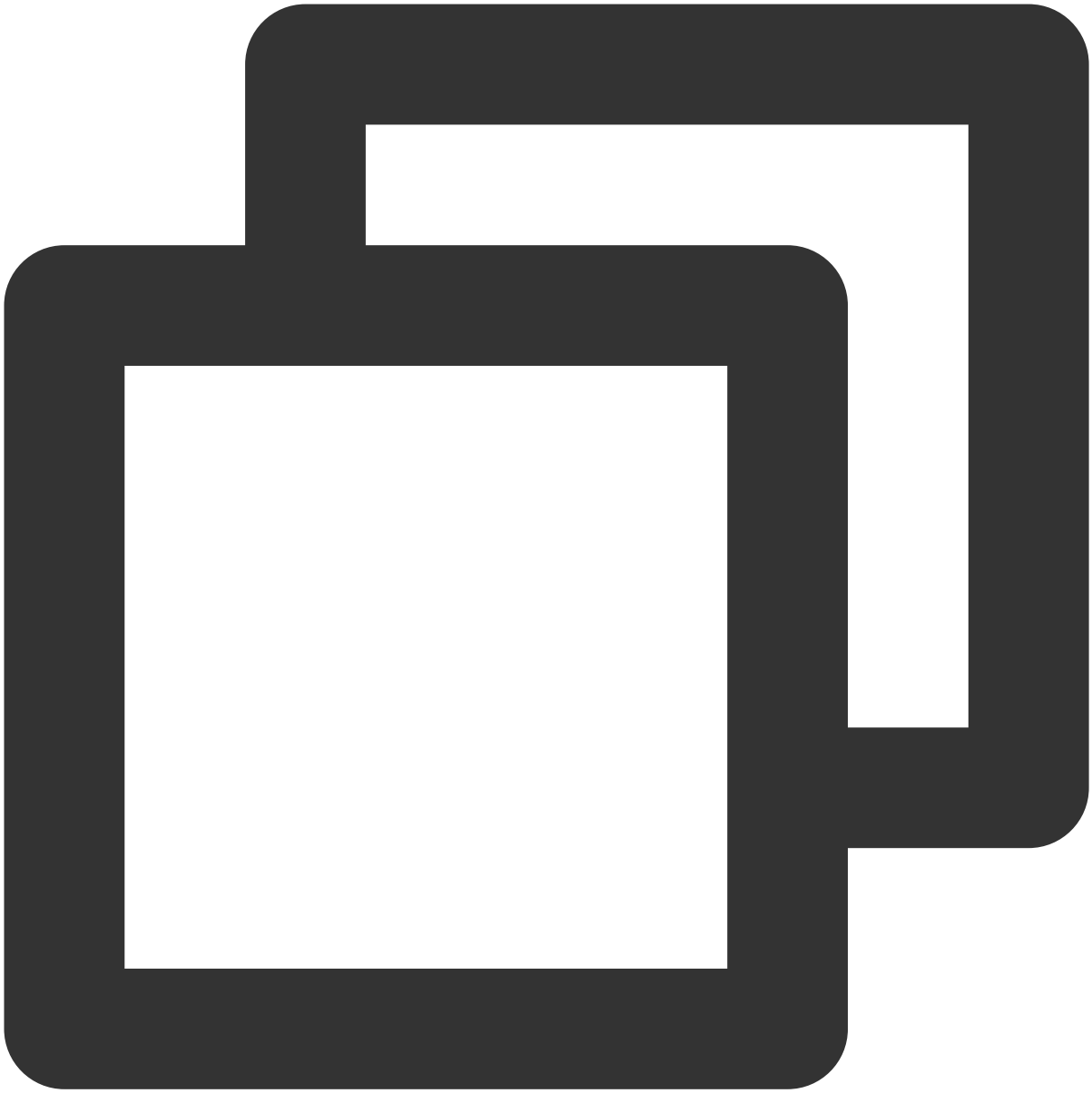
5. To log in to GPU instances, see Log into Linux Instance Using Standard Login Method. You can also use other login methods:
Logging into Linux instances via remote login tools
Logging into Linux instance via SSH key
6. Run the `wget` command to download the installer using the URL copied in Step 4, 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:



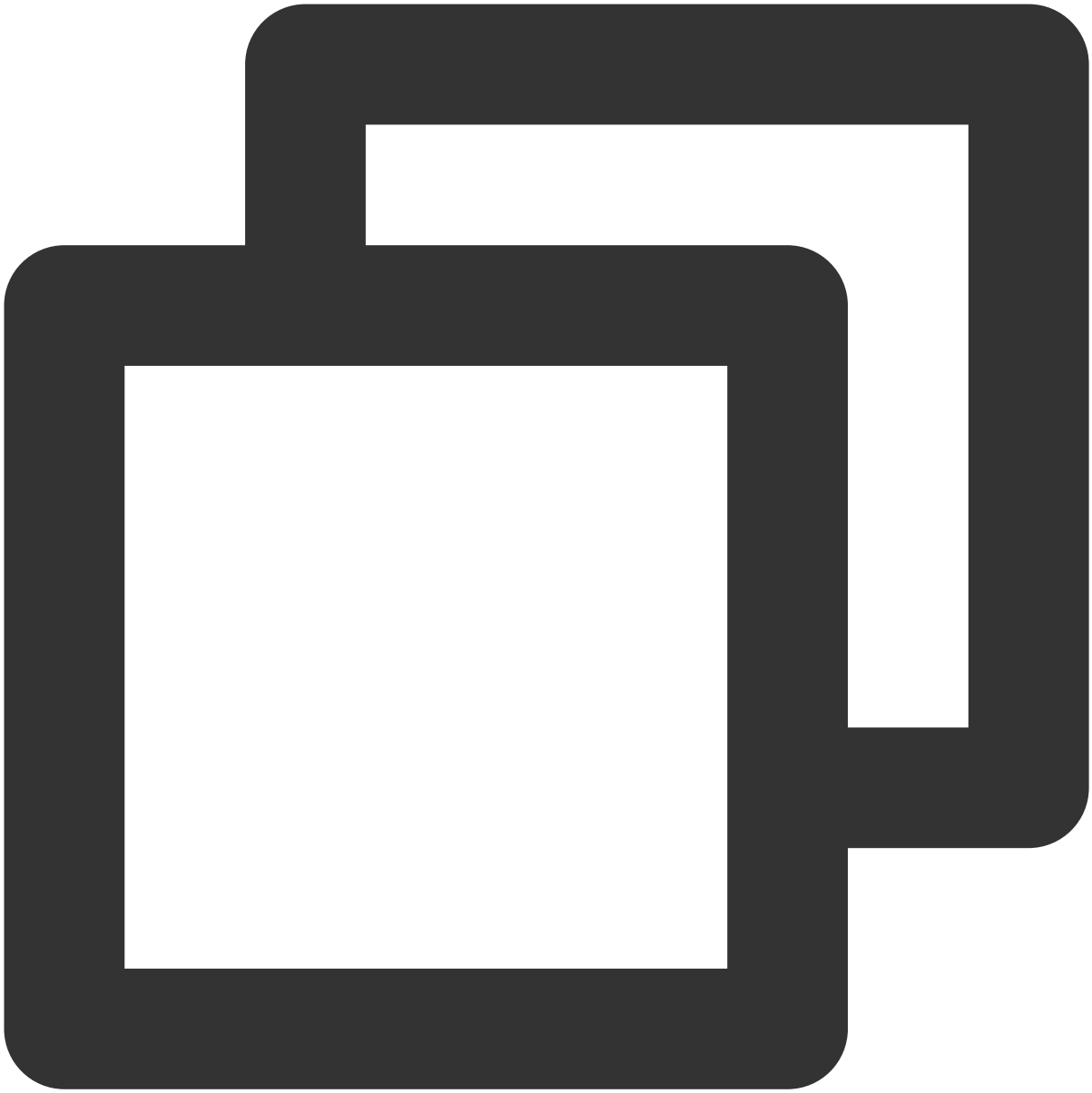
```
sudo chmod +x cuda_10.1.105_418.39_linux.run
```



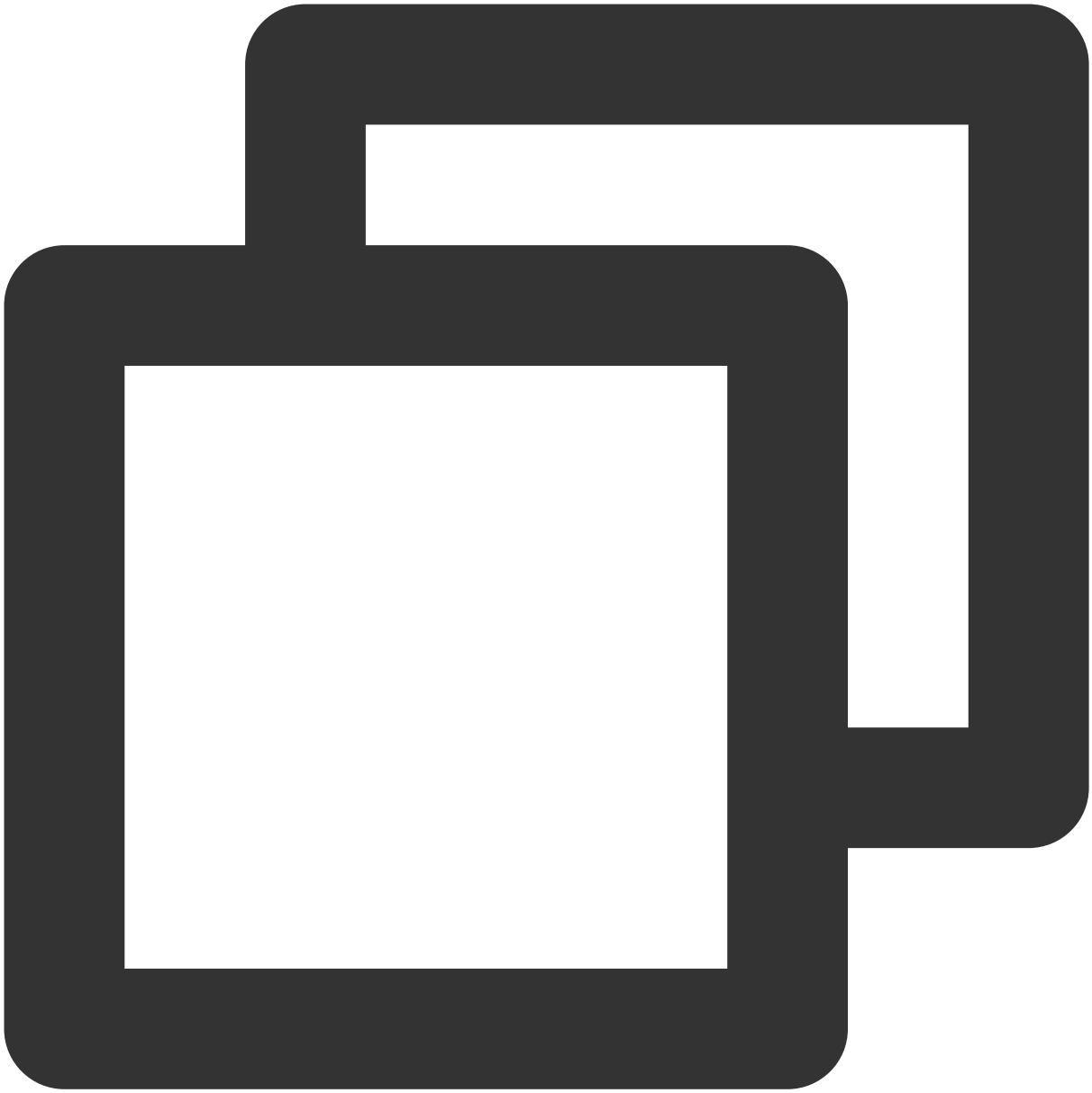
```
./cuda_10.1.105_418.39_linux.run --toolkit --samples --silent
```

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
```

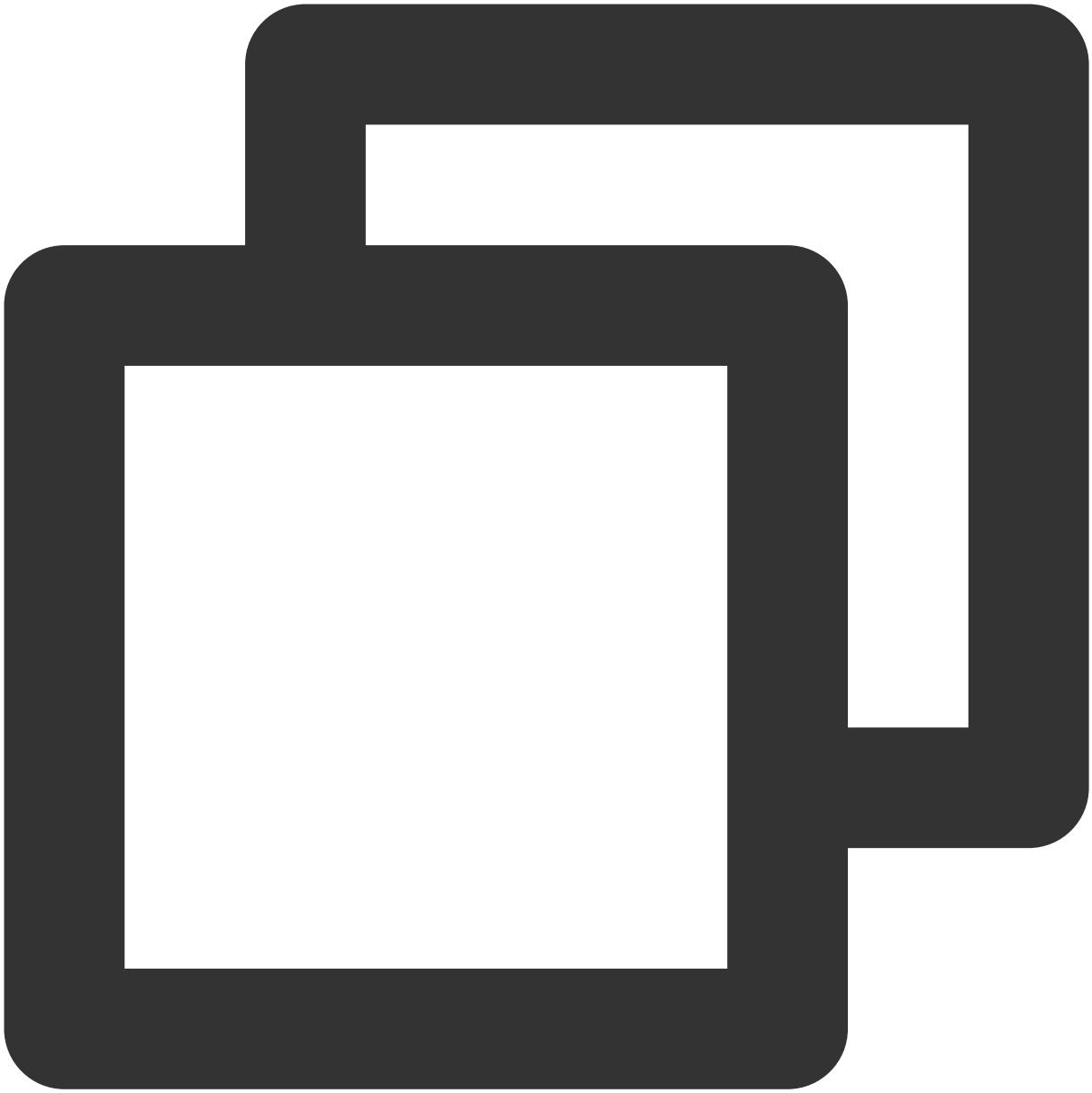


```
source /etc/profile
```

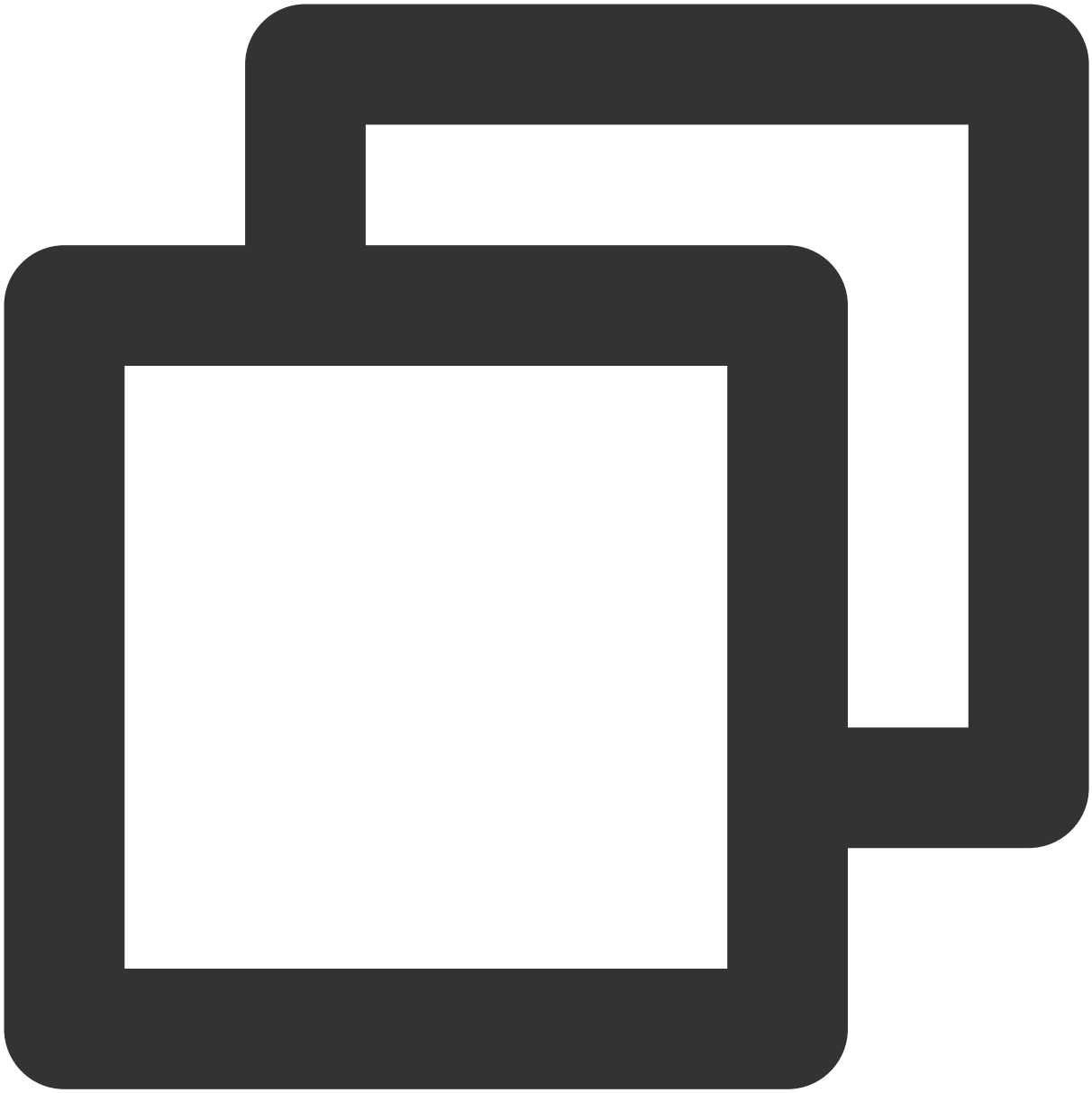
10.

Ru

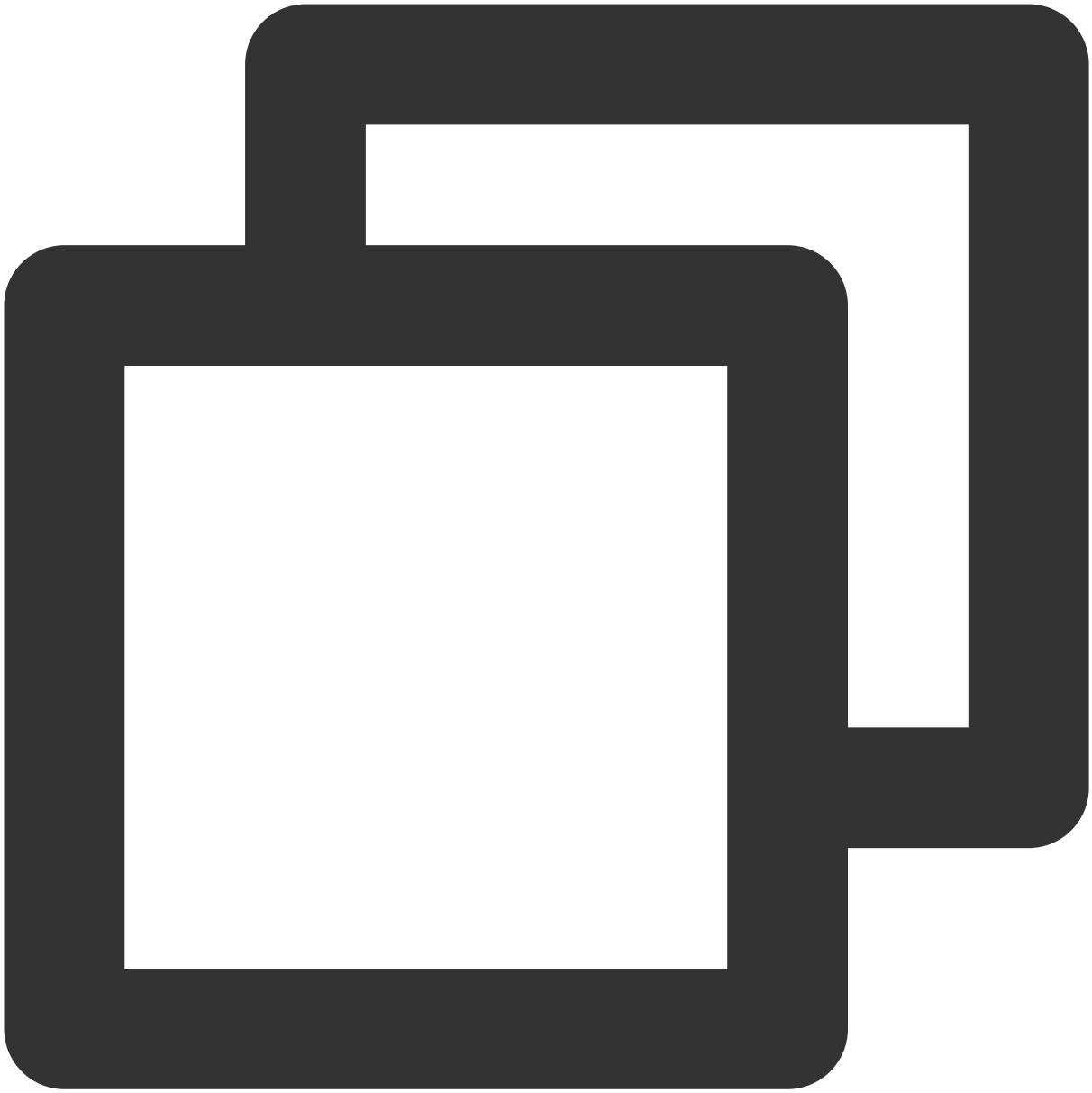
n the following commands in sequence to check whether CUDA Toolkit has been installed:



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



```
make
```



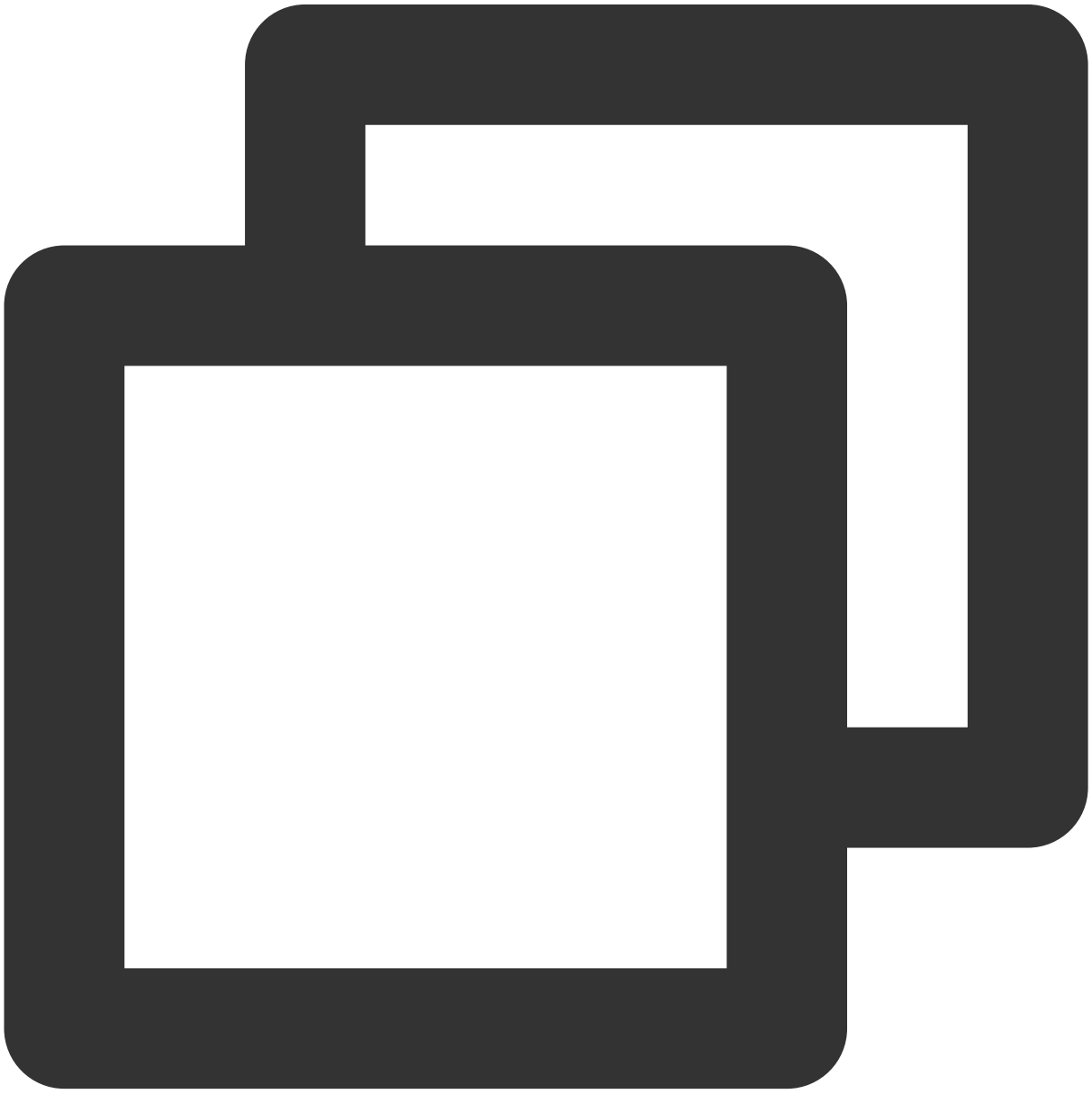
```
./deviceQuery
```

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

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

```
[root@VM-32-212-centos deviceQuery]# make
/usr/local/cuda/bin/nvcc -ccbin g++ -I../../common/inc  -m64    -gencode ar
30,code=sm_30 -gencode arch=compute_35,code=sm_35 -gencode arch=compute_37,
-gencode arch=compute_50,code=sm_50 -gencode arch=compute_52,code=sm_52 -ge
compute_60,code=sm_60 -gencode arch=compute_61,code=sm_61 -gencode arch=com
e=sm_70 -gencode arch=compute_75,code=sm_75 -gencode arch=compute_75,code=c
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:



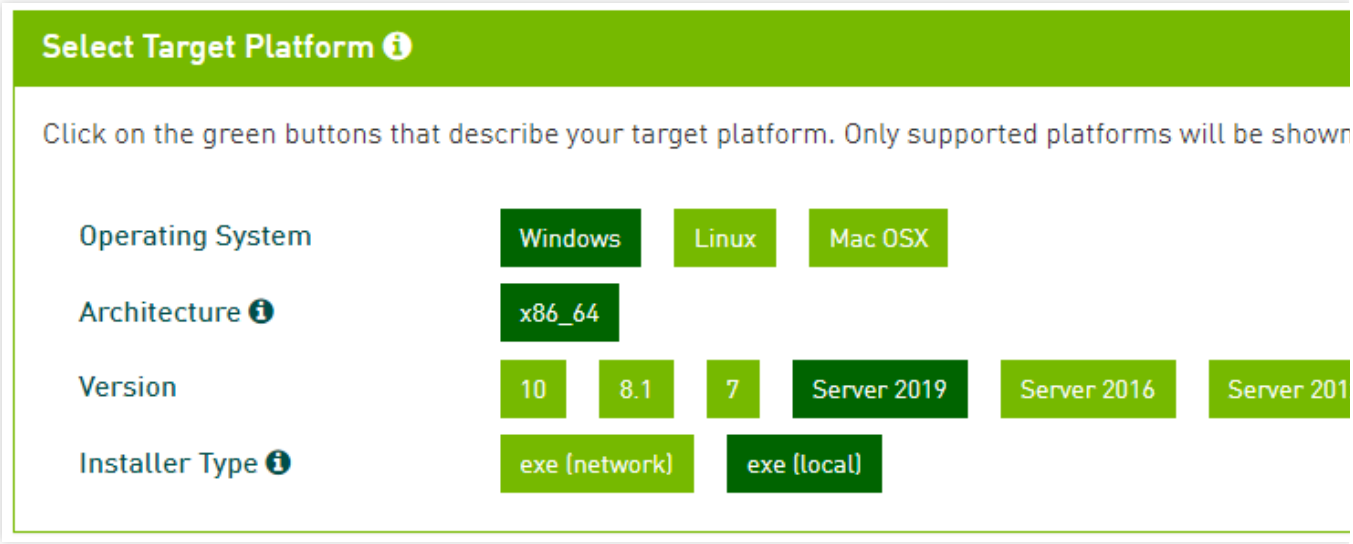
```
yum install -y gcc-c++
```

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.

Upgrading GRID 16 Driver

Last updated：2024-02-06 14:50:17

Background

After July 31, 2023, NVIDIA will no longer provide technical support for the old-version GRID License Server. Additionally, the old-version GRID License purchased by Tencent Cloud will expire on March 30, 2024. To prevent your operations from being impacted by the expiration of the driver, we suggest upgrading your current GRID driver to GRID 16 which is supported by the new-version GRID License Server.

Note:

For more information on NVIDIA's discontinuation of technical support for the old-version GRID License Server, see: NVIDIA Virtual GPU Software License Server End of Life Notice.

Upgrade Plan

The upgrade plan may vary depending on the type of card.

GPU Passthrough Instance Specifications

If your instance is of GNV4v, GNV4, or GN7vw whole card GPU Specifications, you may select one of the following methods for the GRID driver upgrade, based on your actual needs.

Using Self-defined Image for Reinstallation: If you need to retain your operational environment within the original image, we recommend you uninstall the old driver in the original image, upgrade to GRID 16 driver, create it as Self-defined Image and then reinstall the system. The driver upgrade steps will be explained in an internal message.

Replacing the Public Image: If you do not wish to retain your operational environment within the original image, you may opt to employ the 'system reinstallation' feature available on the Cloud GPU Service console<1>in March, choosing to reinstall via a public image pre-installed with GRID 16.2.

Instance Specifications for GPU Partitioning

If your instance happens to be a GNV4v or GN7vw that splits GPU specifications, Tencent Cloud will carry out an upgrade through the Instance Repair Task, following these steps:

1. A one-week advance notification will be issued, or an after-sales manager will get in touch, informing you of the expected update time.
2. In accordance with instructions provided in an internal message, create a self-defined image. If you use the pre-installed GRID 16.2 public image to reinstall the system, you may skip this step.

3. Forty-eight hours before the upgrade, log in to the cloud server console and select **Maintenance Task** > Task List on the left sidebar.
4. On the **Maintenance Task List** page, view the task list for **Instance Maintenance and Upgrades**.
5. Click on **Authorization**/**Appointment** in the pop-up window, perform authorization and time appointment operations to complete the authorization.
6. After the instance maintenance and upgrade task is completed, use the self-defined image created in Step 2 or select the public image pre-installed with GRID 16.2 to reinstall the system and complete the upgrade.

Note:

If you fail to complete the authorization procedure, upgrade will initiate by default 48 hours after the repair task of the instance is assigned.

In case of particular requirements pertaining to the pace of the upgrade, you might consult your Tencent Cloud architect or after-sales manager for support.

Implementing Image Quality Enhancement with GN7vi Instances

Last updated：2024-01-11 17:11:13

Overview

This document describes how to perform video encoding and decoding as well as AI image quality enhancement on GN7vi instances, which are fully compatible with the open-source FFmpeg.

Directions

Preparing the instance environment

Create an instance:

Instance: Select an instance based on the requirements for GPU and the number of cores as instructed in Computing Instance.

Image: Select an image from the available images in the table.

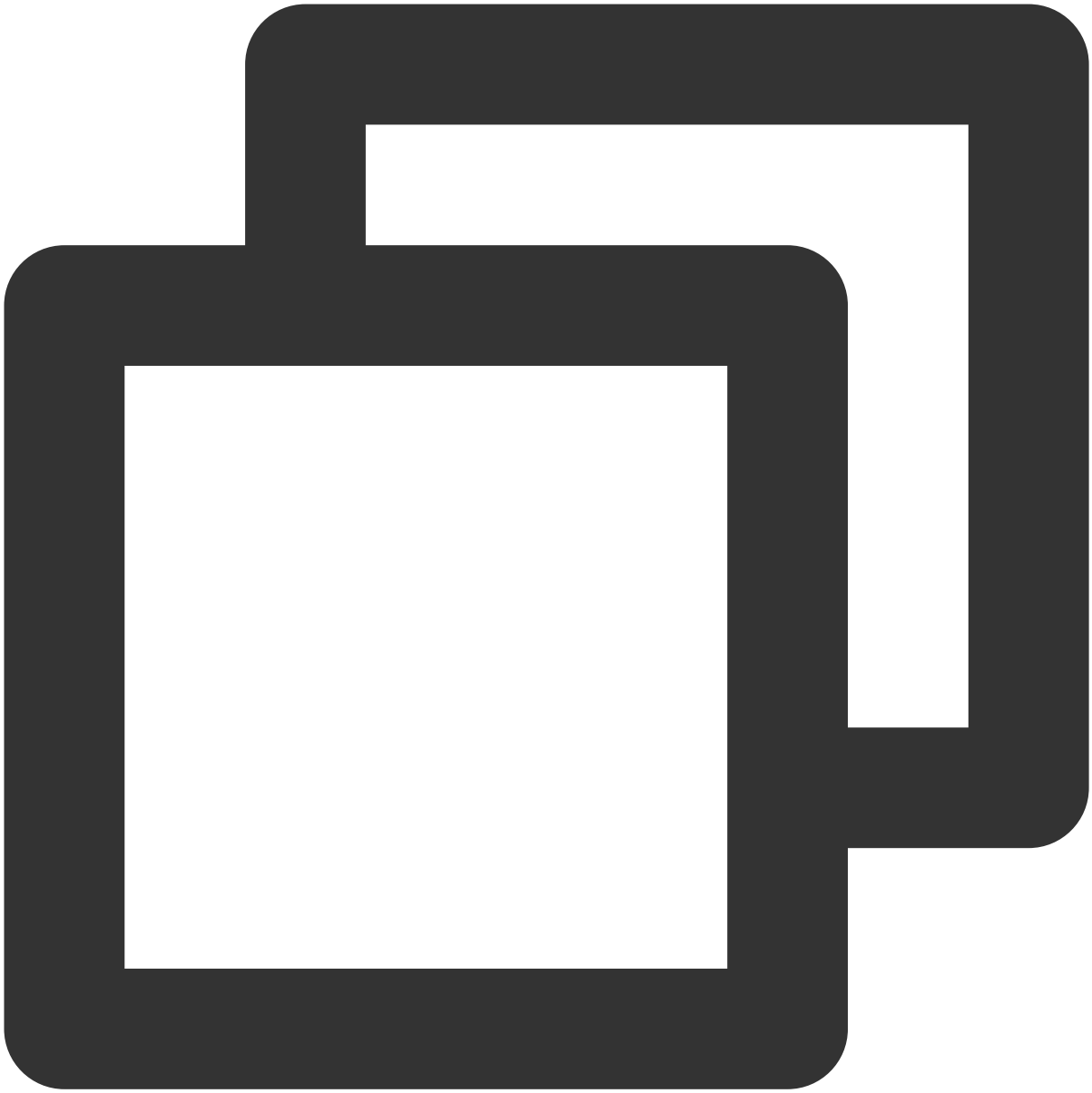
Driver: The automatic installation of CUDA and cuDNN is optional. You can also install them manually after creating the instance as needed:

Log in to the created GPU instance as instructed in Logging in Using Standard Method (Recommended).

Run the following command to have the GPU driver, CUDA, and cuDNN installed automatically.

Note:

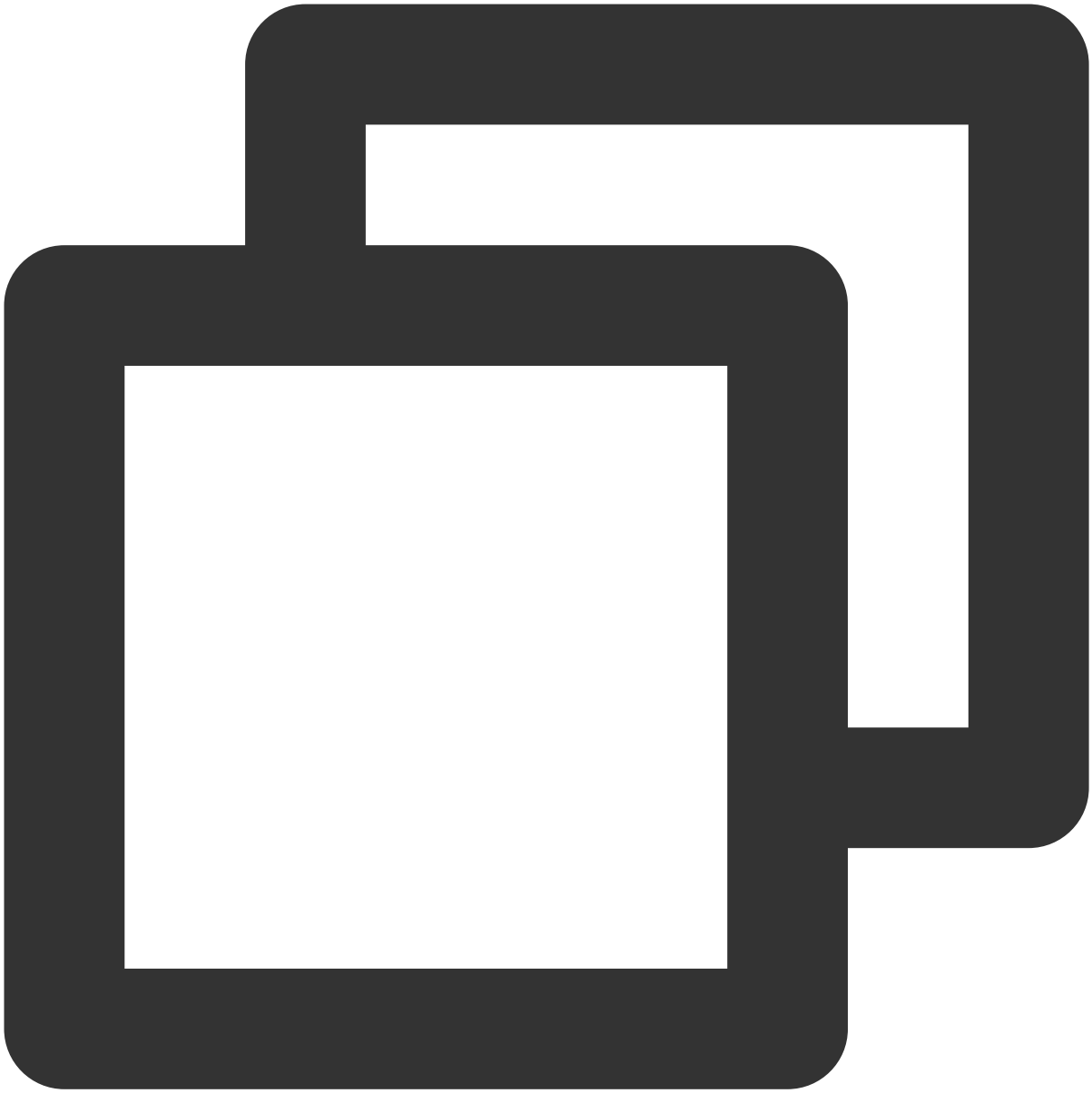
Configure the environment with GPU driver 460.106.00, CUDA 11.2.2, and cuDNN 8.2.1.



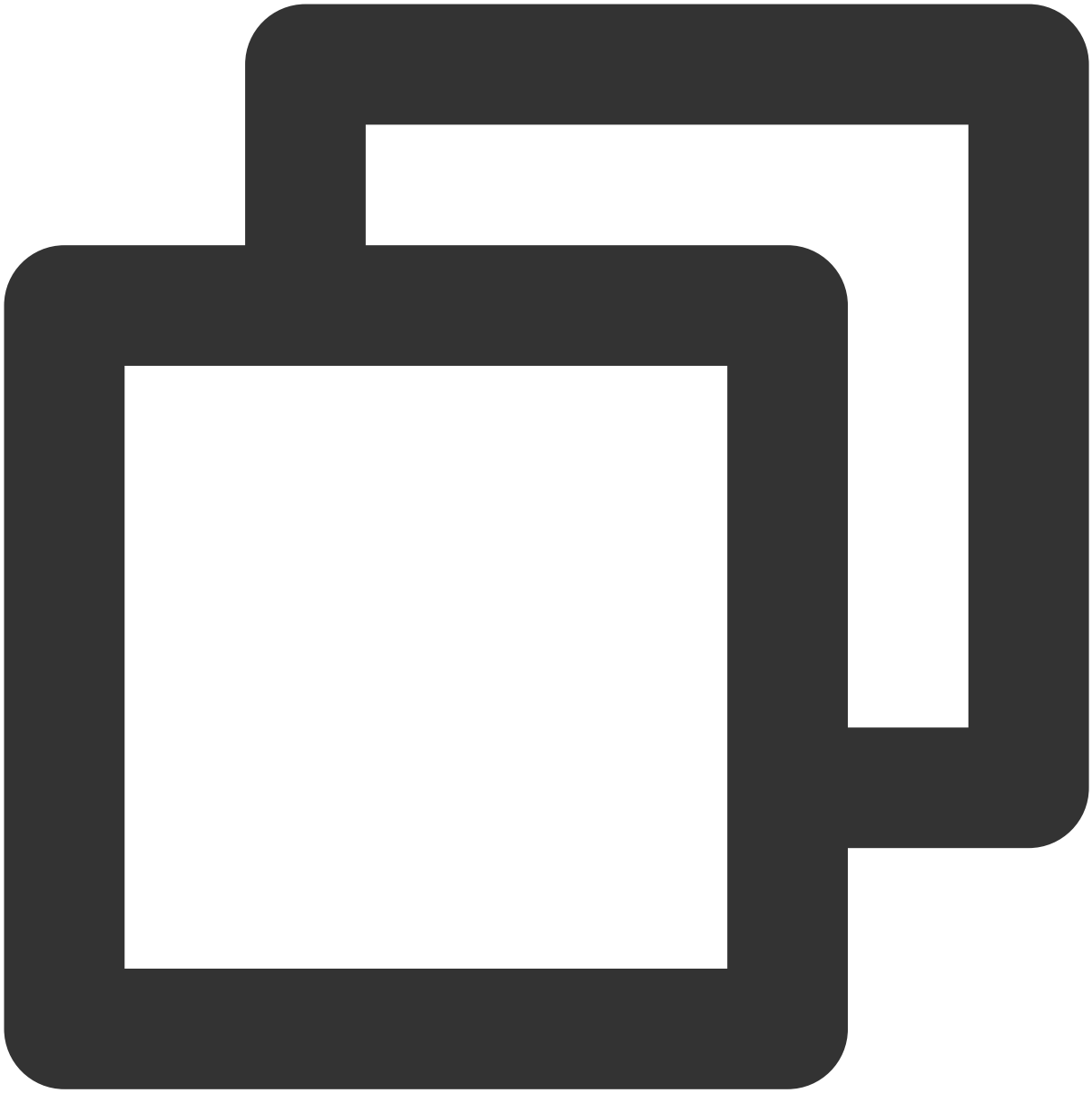
```
wget https://gpu-related-scripts-1251783334.cos.ap-guangzhou.myqcloud.com/gpu-au
```

File overview

Run the following commands in sequence to view all files under `tscsdk-center` .



```
cd /usr/local/qcloud/tscsdk-center
```



```
ls -l
```

The following information appears:

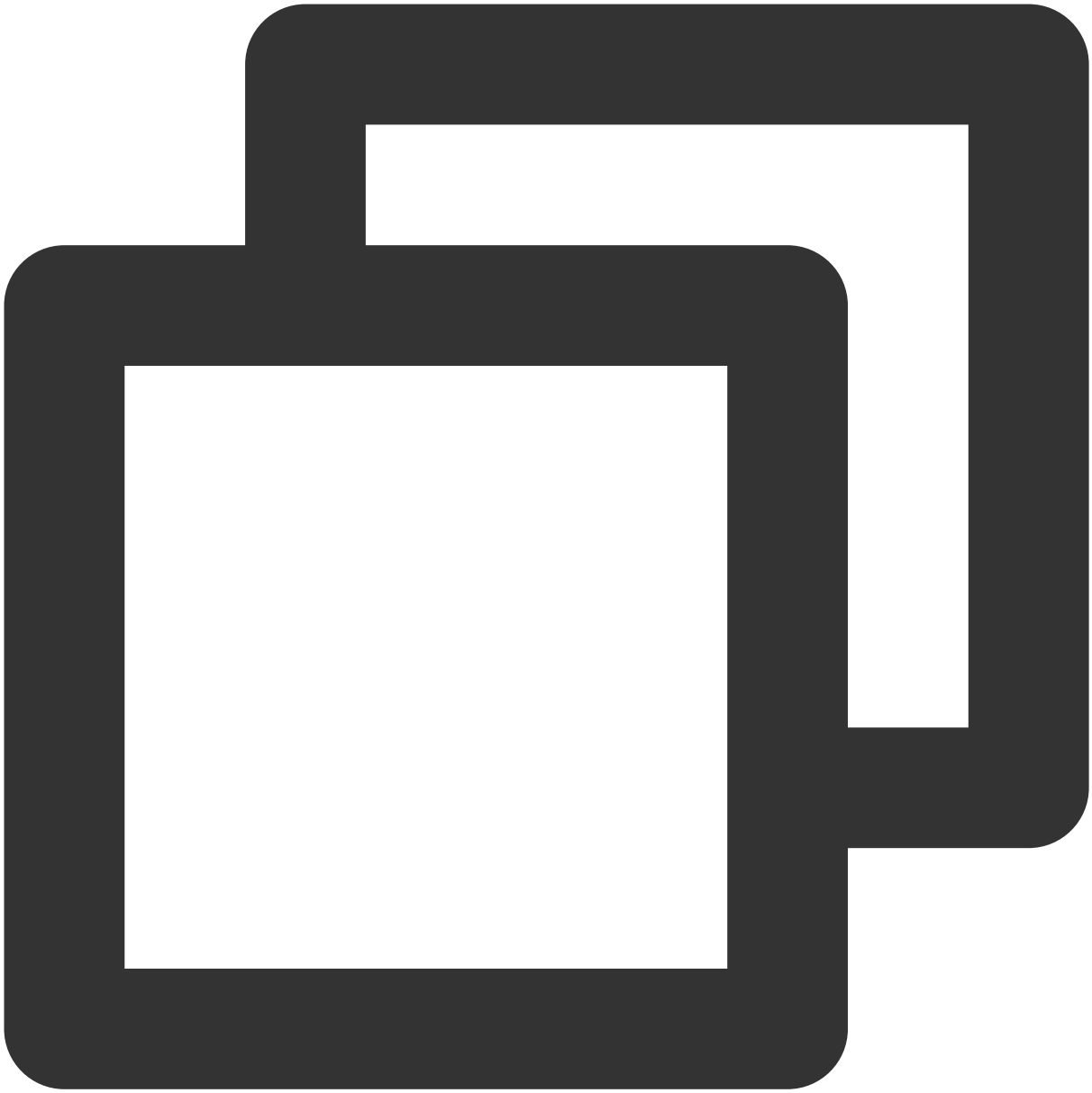
```
drwxr-xr-x 2 root root     4096 Aug 26 10:42 fflib_gpu
-rwxr-xr-x 1 root root 20532592 Aug 26 10:01 ffmpeg
drwxr-xr-x 3 root root     4096 Aug 26 17:55 tenmodel
drwxr-xr-x 2 root root     4096 Aug 29 17:28 videos
```

The description is as follows:

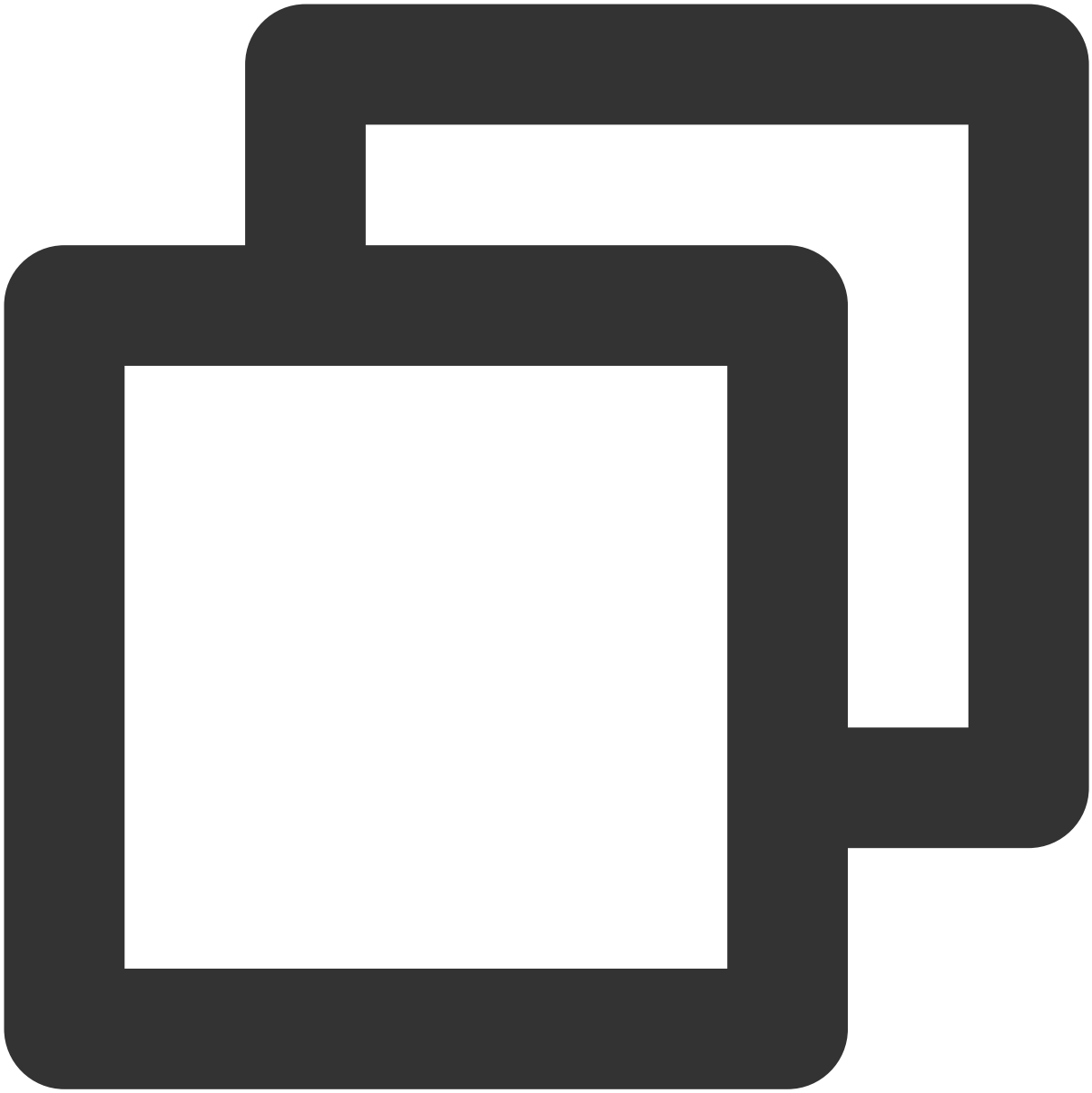
File Name	Description
fflib_gpu	Dependency library for image quality processing.
ffmpeg	FFmpeg program embedded with the image quality processing feature.
tenmodel	AI model used in image quality processing.
videos	Built-in sample videos.

Getting started

1. Run the following commands in sequence to set environment variables.



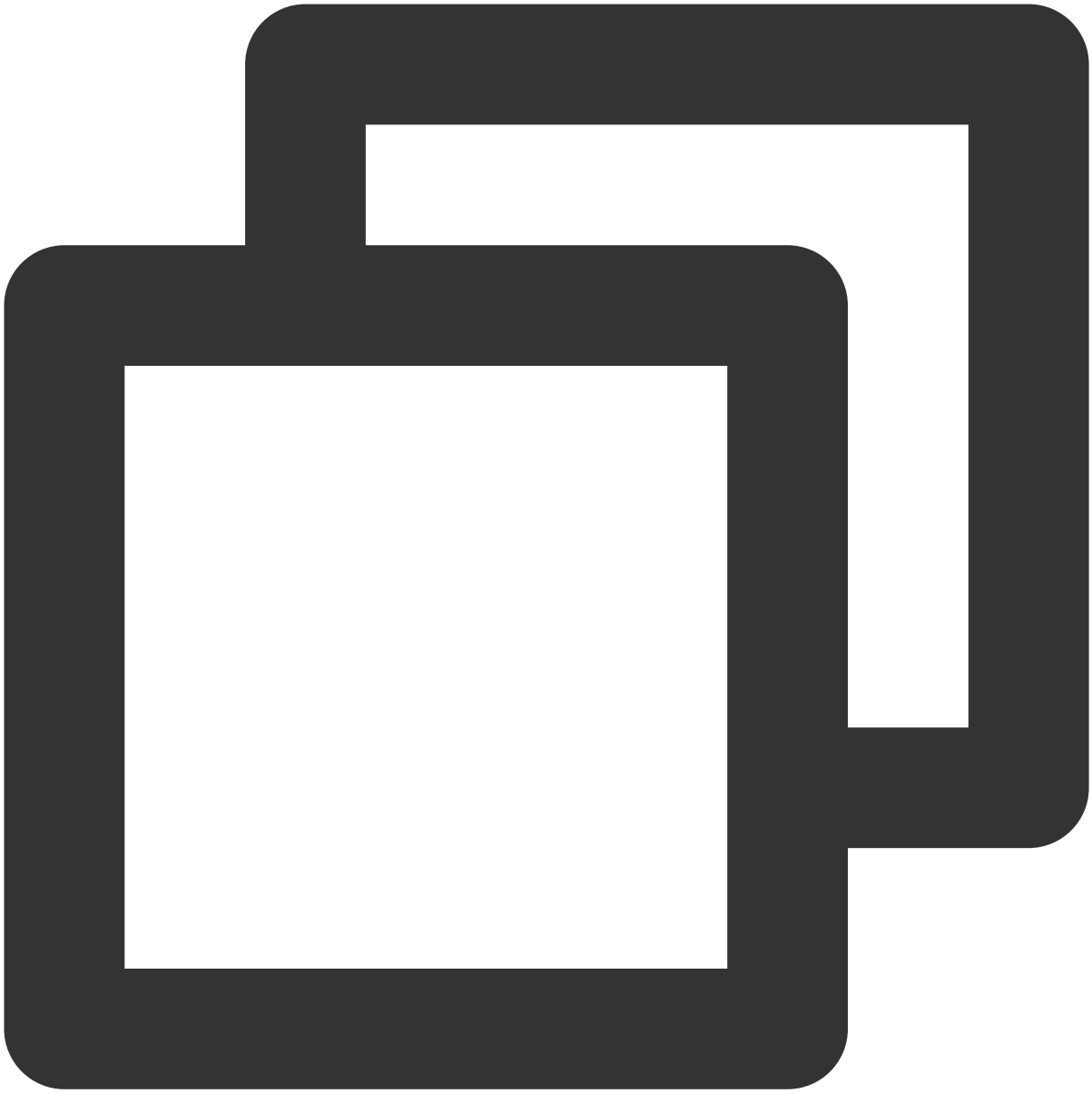
```
cd /usr/local/qcloud/tscsdk-center
```



```
export LD_LIBRARY_PATH=./fflib_gpu:$LD_LIBRARY_PATH
```

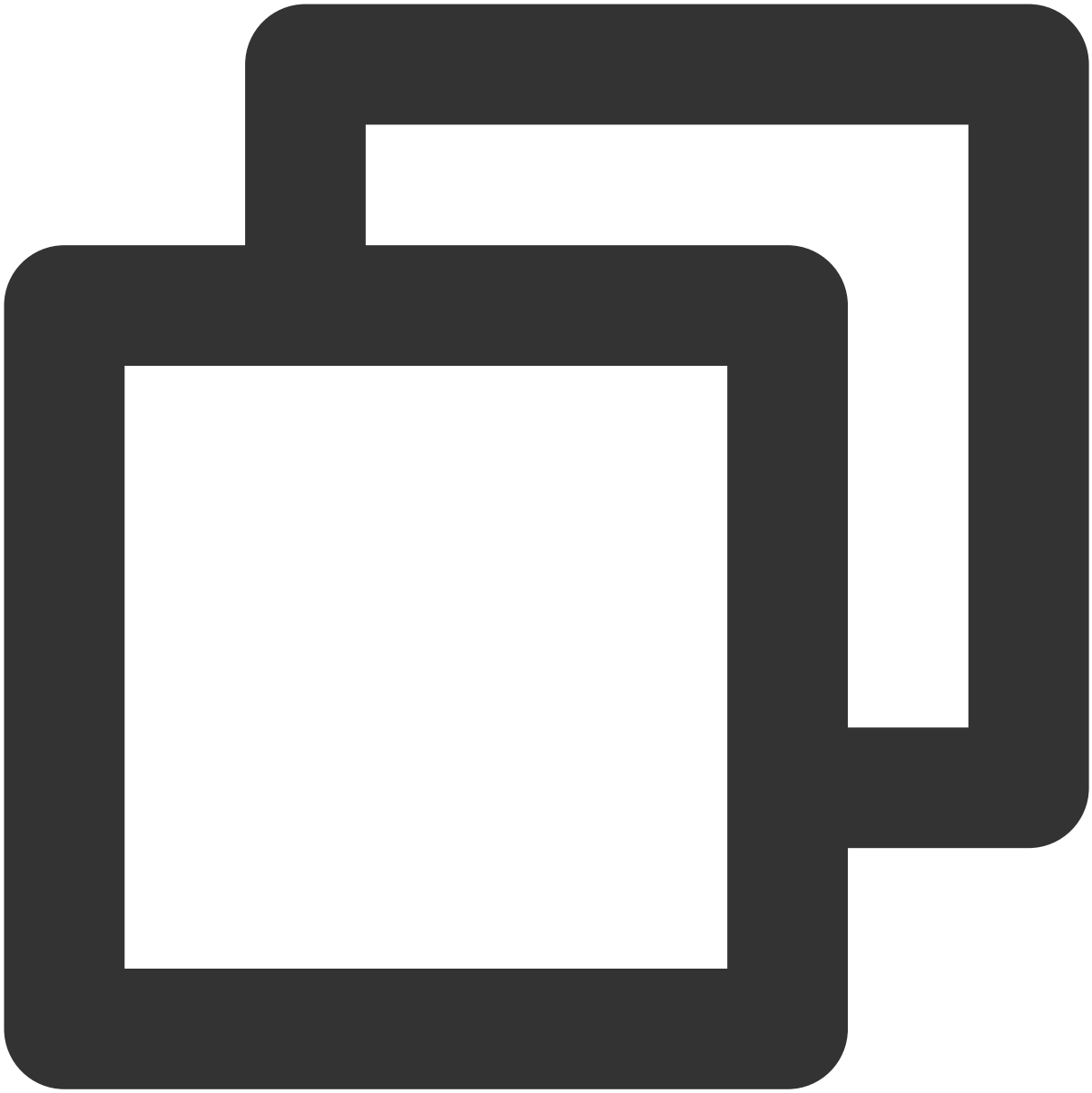
2. Under the `tscsdk-center` directory, run the following commands in sequence to generate the sample output video after image quality processing.

LD video processing: LD videos usually have a resolution of up to 720p. The command uses the standard super resolution model in the balance mode of tenfilter and unsharp sharpening.



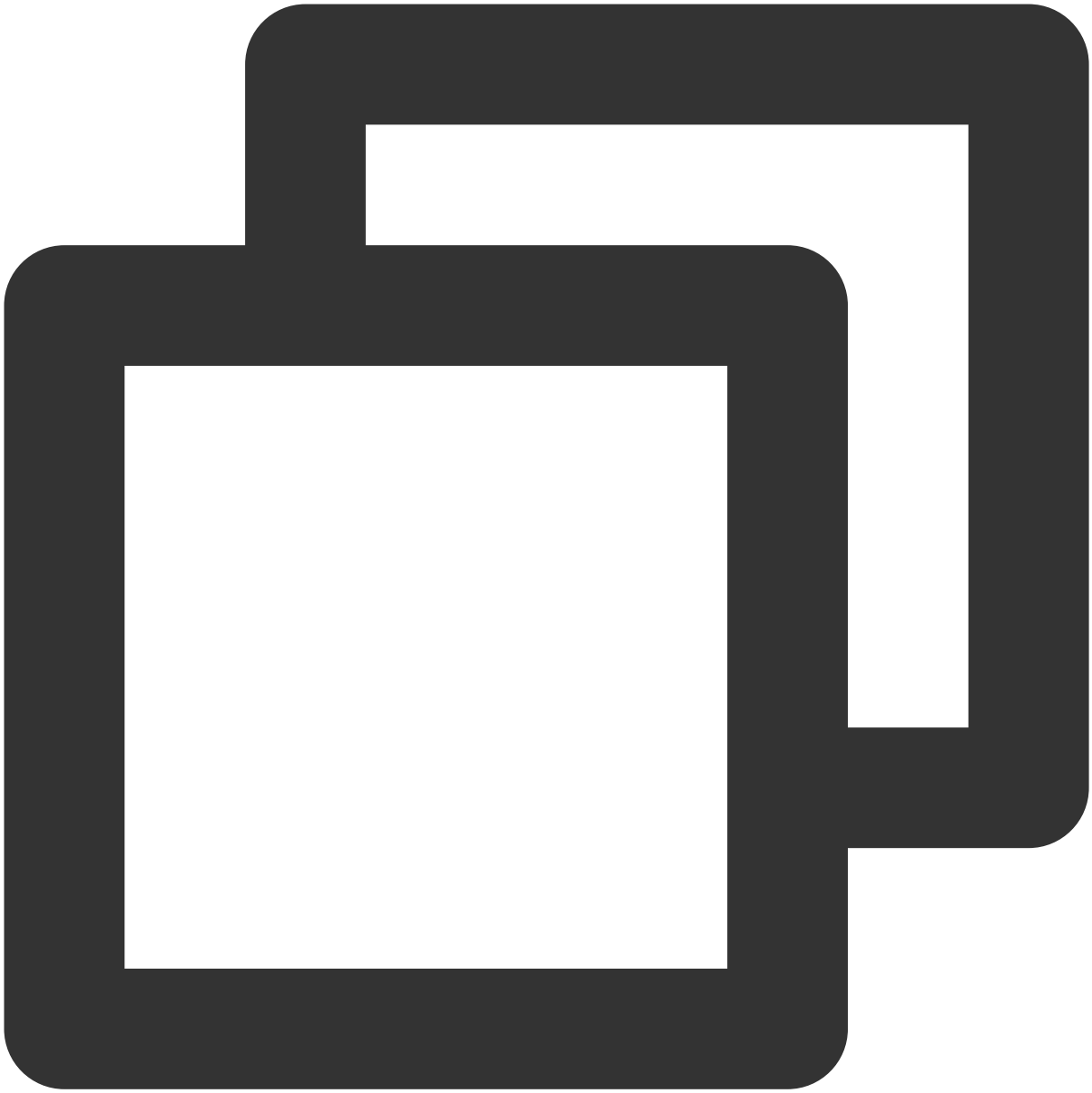
```
./ffmpeg -i ./videos/input1.mp4 -vf tenfilter=mag_filter=1:mag_sr=2:mag_sr_stre=bal
```

HD video processing: HD videos usually have a resolution of above 720p. The command uses the high-quality super resolution model in tenfilter.

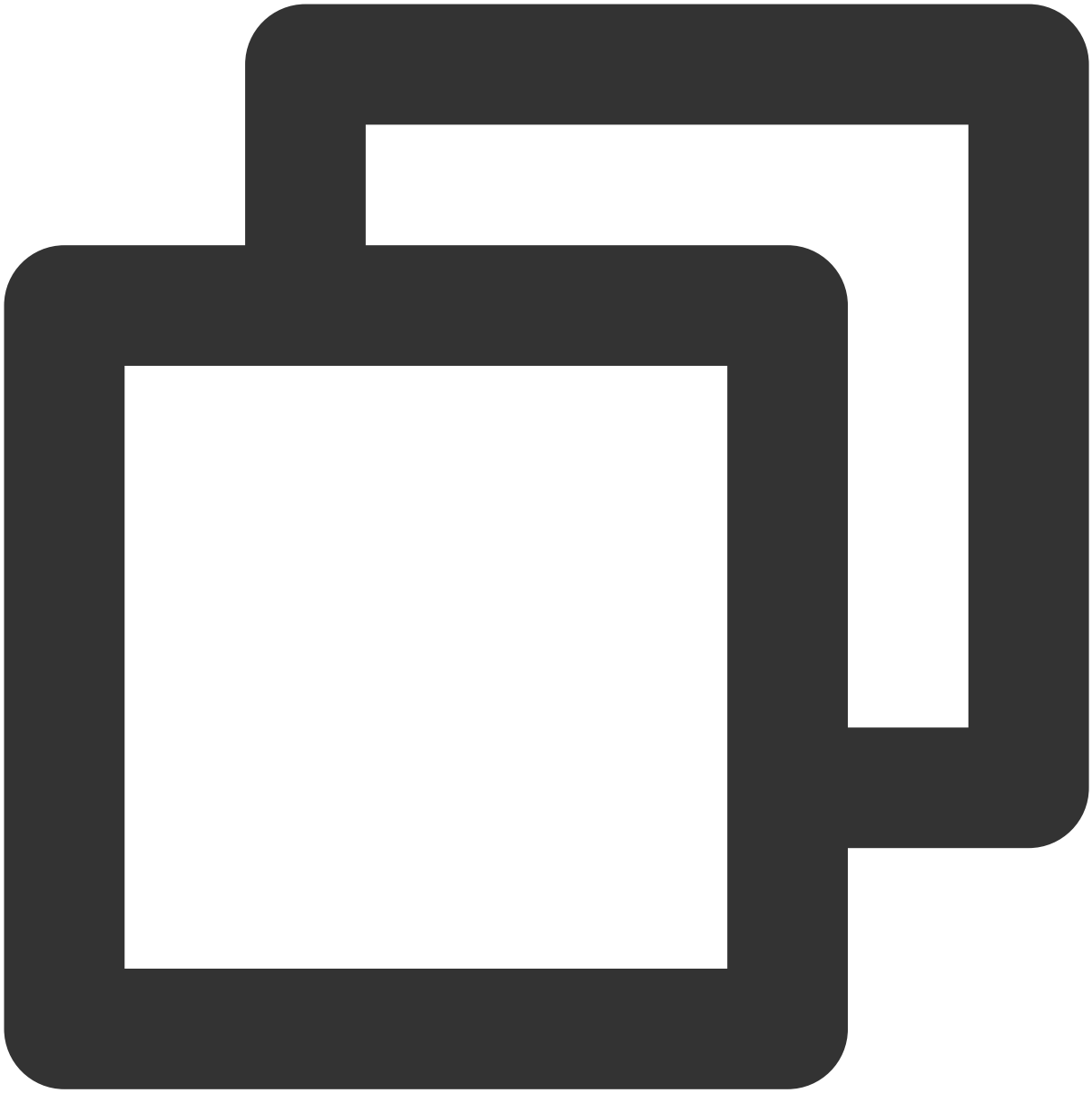


```
./ffmpeg -i ./videos/input2.mp4 -vf tenfilter=mag_srgan=1 -c:v libten264 -ten264opt
```

Fast processing model: The following commands use compression artifact removal in tenfilter, standard super resolution model in general mode, and unsharp sharpening.



```
./ffmpeg -i ./videos/input1.mp4 -vf tenfilter=af=auto,tenfilter=mag_filter=1:mag_sr
```



```
./ffmpeg -i ./videos/input2.mp4 -vf tenfilter=af=auto,tenfilter=mag_filter=1:mag_sr
```

Note:

The running speed of a model is affected by the input resolution. The higher the resolution, the slower the running. When a specific AI model is run for the first time, the initialization will take a long time. During subsequent command execution, the speed will be improved significantly. You can evaluate the running speed by comparing subsequent execution results.

Some parameters in the `ffmpeg` command are as described below:

Parameter	Description

-i videos/input1.mp4	Specifies the input video file.
-vf tenfilter=mag_srgan=1	Specifies the video processing filter graph. For more information on parameter descriptions, see List of features of the AI model for video processing.
-c:v libten264	Specifies Tencent's proprietary Ten264 or Ten265 as the video encoder.
-ten264opts crf=26:vbv-maxrate=2000	Sets video encoder parameters. For more information on parameter descriptions, see List of features of the video encoder.
-y output.mp4	Specifies the output video file to automatically overwrite the existing file.

3. Wait for the program to end and download the output video file. We recommend you use Xshell or MobaXterm. The following are screenshots of the four video files output by the commands.

output1.mp4

fast_output1.mp4

output2.mp4

fast_output2.mp4

Screenshot taken at 01:15 (minute)

Screenshot taken at 01:15 (minute)

Screenshot taken at 00:10 (minute)

Screenshot taken at 00:10 (minute)

Feature List

The tscsdk-center consists of two parts: AI model for video processing and Tencent's proprietary video encoder. The AI model for video processing is integrated by using the FFmpeg filter mechanism, so that filters can embed AI inference capabilities into video encoding/decoding and processing processes. This improves hardware utilization efficiency and throughput. With Tencent's proprietary video encoder, a higher video encoding compression rate is delivered in addition to image quality enhancement.

List of features of the AI model for video processing

The AI model for video processing is integrated in a filter named "tenfilter" and is called and configured through `"-vf tenfilter=name1=value1:name2=value2"` . One AI model can be enabled in one tenfilter, and free combinations are available when there are multiple tenfilters.

All AI models are as described below:

Model or Feature Name	Parameter	Sample

General parameters	mdir: It is the configuration file path of the model, which defaults to `./tenmodel/tve-conf.json`. gpu: It is the GPU No. of the tenfilter.	tenfilter=mdir=./tenmodel/tve-conf.json:gpu=1
Compression artifact removal	af: It is the strength of compression artifact removal, which can be only `auto` currently.	tenfilter=af=auto
Face protection	face_protect_enable: The face protection logic is enabled when it is `1`. face_af_ratio: It is the face area denoising weakening coefficient. face_sp_ratio: It is the face area sharpening coefficient.	tenfilter=face_protect_enable=1:face_af_ratio=0.5:face_sp_rat
Video frame interpolation	mag_fps: Video frame interpolation is enabled when it is `1`. fps: It is the target frame rate.	tenfilter=mag_fps=1:fps=60
Color enhancement	mag_filter: It needs to be set to `1`. cebb: Color enhancement is enabled when it is `1`.	tenfilter=mag_filter=1:cebb=1
Standard super resolution	mag_filter: It needs to be set to `1`. mag_sr: It is the super resolution rate.	tenfilter=mag_filter=1:mag_sr=2:mag_sr_stre=normal

	Currently, only the twice super resolution is supported. mag_sr_stre: It is the super resolution mode, which can be set to `normal` or `balance`.	
High-quality super resolution	mag_srgan: High-quality super resolution is enabled when it is `1`.	tenfilter=mag_srgan=1
Video noise removal	mag_filter: It needs to be set to `1`. dn: It is the noise removal strength, which can be only `3` currently.	tenfilter=mag_filter=1:dn=3
Video image quality enhancementFace enhancementFont enhancement(Support for multiple models)	mag_filter: It needs to be set to `1`. eh: Image quality enhancement is enabled when it is `1`. faceeh: Face enhancement is enabled when it is `1`. fonteh: Font enhancement is enabled when it is `1`. prior: It is the priority for executing AI models, for example, "faceeh-eh-fonteh". Corresponding models need to be enabled. When `-parally` is added, parallel optimization is enabled.	Single model:tenfilter=mag_filter=1:eh=1 Multiple models:tenfilter=mag_filter=1:eh=1:faceeh=1:prior=faceeh-parally

Tencent's proprietary video encoder

tscsdk-center contains Ten264 and Ten265 video encoders independently developed by Tencent. Encoder types and parameters can be set through command parameters during video processing.
Each encoder can be specified and set in the following ways:

Encoder Name	Method to Specify	Method to Set
Ten264	-vcodec libten264-c:v libten264	-ten264opts name1=value1:name2=value2
Ten265	-vcodec libten265-c:v libten265	-ten265-params name1=value1:name2=value2

The parameters of each encoder are as detailed below:
Ten264 encoder
Ten265 encoder

Parameter	Description
preset	Specifies the configuration of the encoder's encoding parameter set.0: Ultrafast; 1: Superfast; 2: Very fast; 3: Faster; 4: Fast; 5: Medium; 6: Slow; 7: Slower; 8: Very slow; 9: Placebo.
bitrate	Bitrate of the output video in ABR mode.
crf	CRF value in CRF mode.
aq-mode	0: Disable aqmode; 1: Enable aqmode; 2: Variance-based aqmode; 3: Variance-based aqmode towards dark scenes.2 is the default value and produces better SSIM results.
vbv-maxrate	Maximum VBV bitrate. This value is the same as the configured bitrate by default.
vbv-bufsize	VBV buffer size. This value is four times the configured bitrate by default.
rc-lookahead	Length of the lookahead.
scenecut	Whether to enable scene switch. It is enabled by default and we generally recommend you keep it enabled.
keyint	Maximum keyframe interval. It is 256 by default and can be configured as needed; generally, you should configure it as the number of frames with a time interval of 2–5s.
threads	Number of threads in the used thread pool.
lookahead-threads	Number of threads used for lookahead.

profile	"baseline", "main", "high", "high422", and "high444".

Parameter	Description
preset	Specifies the configuration of the encoder's encoding parameter set.-1: Ripping; 0: Placebo; 1: Very slow; 2: Slower; 3: Slow; 4: Universal; 5: Medium; 6: Fast; 7: Faster; 8: Very fast; 9: Superfast.
rc	Bitrate control method.0: CQP; 1: ABR_VBV; 2: ABR; 3: CRF_VBV; 4: CRF.
bitrate	Bitrate of the output video in ABR mode.
crf	CRF value in CRF mode. Value range: [1,51].
aq-mode	0: Disable aqmode; 1: Enable aqmode; 2: Variance-based aqmode; 3: Variance-based aqmode towards dark scenes.2 is the default value and produces better SSIM results.
vbv-maxrate	Maximum VBV bitrate. This value is the same as the configured bitrate by default.
vbv-bufsize	VBV buffer size. This value is four times the configured bitrate by default.
rc-lookahead	Length of the lookahead.
scenecut	Scene switch threshold. Value range: [0,100]. 0 indicates disabled. It is enabled by default and we generally recommend you keep it enabled.
open-gop	Whether to enable open GOP.0: Disabled; 1: Enabled. It is enabled by default; in order to support random access in the live streaming scenario, we recommend you disable it.
keyint	Maximum keyframe interval. It is 256 by default and can be configured as needed; it needs to be a multiple of 8 greater than 50.
ltr	Whether to support long-term reference frames. 0: Disabled; 1: Enabled. It is enabled by default; if the hardware device that plays back HEVC videos is poor, we recommend you disable it.
pool-threads	Number of threads in the thread pool used by WPP. It is the same as the number of CPU cores by default; if you want to reduce CPU usage, lower this number.

Usage Recommendations

tscsdk-center allows for the flexible control of each AI model. If you have special requirements or scenarios, you can set the switch for each model and combine different models to deliver better video processing effects.
tscsdk-center provides two super resolution models. The standard model is suitable for earlier sources at a low resolution, while the high-quality model is more suitable for HD sources. We recommend you evaluate the effects of the two models while considering the video source type.
In tscsdk-center, AI models need to run on GPU, while video encoders run only on CPU. In most cases, when the GPU computing power is fully used, there will be some idle CPU space. Therefore, some video processing tasks running only on CPU can be assigned, such as video transcoding, to fully utilize hardware resources.