

# tf.test.is\_gpu\_available



View source on [GitHub](https://github.com/tensorflow/tensorflow/blob/v2.16.1/tensorflow/python/framework/test_util.py#L2058-L2116) ([https://github.com/tensorflow/tensorflow/blob/v2.16.1/tensorflow/python/framework/test\\_util.py#L2058-L2116](https://github.com/tensorflow/tensorflow/blob/v2.16.1/tensorflow/python/framework/test_util.py#L2058-L2116))

Returns whether TensorFlow can access a GPU. (deprecated)

**+ View aliases**

## Compat aliases for migration

See [Migration guide](https://www.tensorflow.org/guide/migrate) (<https://www.tensorflow.org/guide/migrate>) for more details.

### tf.compat.v1.test.is\_gpu\_available

([https://www.tensorflow.org/api\\_docs/python/tf/test/is\\_gpu\\_available](https://www.tensorflow.org/api_docs/python/tf/test/is_gpu_available))

```
tf.test.is_gpu_available(  
    cuda_only: bool = False,  
    min_cuda_compute_capability: Optional[tuple[int, int]] = None  
) -> bool
```

**Deprecated:** THIS FUNCTION IS DEPRECATED. It will be removed in a future version. Instructions for updating: Use `tf.config.list_physical_devices('GPU')` instead.

**Warning:** if a non-GPU version of the package is installed, the function would also return False. Use

### tf.test.is\_built\_with\_cuda

([https://www.tensorflow.org/api\\_docs/python/tf/test/is\\_built\\_with\\_cuda](https://www.tensorflow.org/api_docs/python/tf/test/is_built_with_cuda)) to validate if TensorFlow was build with CUDA support.

For example,

```
>>> gpu_available = tf.test.is_gpu_available()  
>>> is_cuda_gpu_available = tf.test.is_gpu_available(cuda_only=True)  
>>> is_cuda_gpu_min_3 = tf.test.is_gpu_available(True, (3,0))
```

---

## Args

---

<b>cuda_only</b>	limit the search to CUDA GPUs.
<b>min_cuda_compute_capability</b>	a (major,minor) pair that indicates the minimum CUDA compute capability required, or None if no requirement.

---

Note that the keyword arg name "cuda\_only" is misleading (since routine will return true when a GPU device is available irrespective of whether TF was built with CUDA support or ROCm support. However no changes here because

++ Changing the name "cuda\_only" to something more generic would break backward compatibility

++ Adding an equivalent "rocm\_only" would require the implementation check the build type. This in turn would require doing the same for CUDA and thus potentially break backward compatibility

++ Adding a new "cuda\_or\_rocm\_only" would not break backward compatibility, but would require most (if not all) callers to update the call to use "cuda\_or\_rocm\_only" instead of "cuda\_only"

---

## Returns

True if a GPU device of the requested kind is available.

---

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates. Some content is licensed under the [numpy license](https://numpy.org/doc/stable/license.html) (<https://numpy.org/doc/stable/license.html>).

Last updated 2024-03-27 UTC.