Module: tf.compat.v1.config

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/config#top_of_page)
* [Modules](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/config#modules)
* [Functions](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/config#functions)

Public API for tf.config namespace.

Modules

[experimental](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/config/experimental) module: Public API for tf.config.experimental namespace.

[optimizer](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/config/optimizer) module: Public API for tf.config.optimizer namespace.

[threading](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/config/threading) module: Public API for tf.config.threading namespace.

Functions

[experimental\_connect\_to\_host(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_connect_to_host): Connects to a single machine to enable remote execution on it.

[experimental\_list\_devices(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_list_devices): List the names of the available devices.

[experimental\_run\_functions\_eagerly(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_run_functions_eagerly): Enables / disables eager execution of [tf.function](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/function)s.

[get\_soft\_device\_placement(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/get_soft_device_placement): Get if soft device placement is enabled.

[set\_soft\_device\_placement(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/set_soft_device_placement): Set if soft device placement is enabled.

# tf.config.experimental\_connect\_to\_host

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_connect_to_host#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_connect_to_host#aliases)

Connects to a single machine to enable remote execution on it.

### Aliases:

* tf.compat.v1.config.experimental\_connect\_to\_host
* tf.compat.v2.config.experimental\_connect\_to\_host
* tf.config.experimental\_connect\_to\_host

tf.config.experimental\_connect\_to\_host(  
    remote\_host=None,  
    job\_name='worker'  
)

Defined in [python/eager/remote.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/eager/remote.py).

Will make devices on the remote host available to use. Note that calling this more than once will work, but will invalidate any tensor handles on the old remote devices.

Using the default job\_name of worker, you can schedule ops to run remotely as follows:

# Enable eager execution, and connect to the remote host.  
tf.compat.v1.enable\_eager\_execution()  
tf.contrib.eager.connect\_to\_remote\_host("exampleaddr.com:9876")  
  
with ops.device("job:worker/replica:0/task:1/device:CPU:0"):  
  # The following tensors should be resident on the remote device, and the op  
  # will also execute remotely.  
  x1 = array\_ops.ones([2, 2])  
  x2 = array\_ops.ones([2, 2])  
  y = math\_ops.matmul(x1, x2)

#### Args:

* **remote\_host**: a single or a list the remote server addr in host-port format.
* **job\_name**: The job name under which the new server will be accessible.

#### Raises:

* **ValueError**: if remote\_host is None.

# tf.config.experimental\_list\_devices

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_list_devices#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_list_devices#aliases)

List the names of the available devices.

### Aliases:

* tf.compat.v1.config.experimental\_list\_devices
* tf.compat.v2.config.experimental\_list\_devices
* tf.config.experimental\_list\_devices

tf.config.experimental\_list\_devices()

Defined in [python/eager/context.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/eager/context.py).

#### Returns:

Names of the available devices, as a list.

# tf.config.experimental\_run\_functions\_eagerly

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_run_functions_eagerly#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental_run_functions_eagerly#aliases)

Enables / disables eager execution of [tf.function](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/function)s.

### Aliases:

* tf.compat.v1.config.experimental\_run\_functions\_eagerly
* tf.compat.v2.config.experimental\_run\_functions\_eagerly
* tf.config.experimental\_run\_functions\_eagerly

tf.config.experimental\_run\_functions\_eagerly(run\_eagerly)

Defined in [python/eager/def\_function.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/eager/def_function.py).

After calling tf.config.experimental\_run\_functions\_eagerly(True) all invocations of tf.function will run eagerly instead of running through a graph function.

This can be useful for debugging or profiling.

Similarly, calling tf.config.experimental\_run\_functions\_eagerly(False) will revert the behavior of all functions to graph functions.

#### Args:

* **run\_eagerly**: Boolean. Whether to run functions eagerly.

# tf.config.get\_soft\_device\_placement

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/get_soft_device_placement#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/get_soft_device_placement#aliases)

Get if soft device placement is enabled.

### Aliases:

* tf.compat.v1.config.get\_soft\_device\_placement
* tf.compat.v2.config.get\_soft\_device\_placement
* tf.config.get\_soft\_device\_placement

tf.config.get\_soft\_device\_placement()

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

If enabled, an op will be placed on CPU if any of the following are true 1. there's no GPU implementation for the OP 2. no GPU devices are known or registered 3. need to co-locate with reftype input(s) which are from CPU

#### Returns:

If soft placement is enabled.

# tf.config.set\_soft\_device\_placement

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/set_soft_device_placement#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/set_soft_device_placement#aliases)
* [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/set_soft_device_placement#used_in_the_guide)

Set if soft device placement is enabled.

### Aliases:

* tf.compat.v1.config.set\_soft\_device\_placement
* tf.compat.v2.config.set\_soft\_device\_placement
* tf.config.set\_soft\_device\_placement

tf.config.set\_soft\_device\_placement(enabled)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

### Used in the guide:

* [Using GPUs](https://www.tensorflow.org/beta/guide/using_gpu)

If enabled, an op will be placed on CPU if any of the following are true 1. there's no GPU implementation for the OP 2. no GPU devices are known or registered 3. need to co-locate with reftype input(s) which are from CPU

#### Args:

* **enabled**: Whether to enable soft placement.

Module: tf.config.experimental

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental#top_of_page)
* [Classes](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental#classes)
* [Functions](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental#functions)

Public API for tf.config.experimental namespace.

Classes

[class VirtualDeviceConfiguration](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/VirtualDeviceConfiguration): Configuration class for virtual devices for a PhysicalDevice.

Functions

[get\_device\_policy(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_device_policy): Gets the current device policy.

[get\_memory\_growth(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_memory_growth): Get if memory growth is enabled for a PhysicalDevice.

[get\_synchronous\_execution(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_synchronous_execution): Gets whether operations are executed synchronously or asynchronously.

[get\_virtual\_device\_configuration(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_virtual_device_configuration): Get the virtual device configuration for a PhysicalDevice.

[get\_visible\_devices(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_visible_devices): Get the list of visible physical devices.

[list\_logical\_devices(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/list_logical_devices): Return a list of logical devices created by runtime.

[list\_physical\_devices(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/list_physical_devices): Return a list of physical devices visible to the runtime.

[set\_device\_policy(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_device_policy): Sets the current thread device policy.

[set\_memory\_growth(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_memory_growth): Set if memory growth should be enabled for a PhysicalDevice.

[set\_synchronous\_execution(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_synchronous_execution): Specifies whether operations are executed synchronously or asynchronously.

[set\_virtual\_device\_configuration(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_virtual_device_configuration): Set the virtual device configuration for a PhysicalDevice.

[set\_visible\_devices(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_visible_devices): Set the list of visible devices.

# tf.config.experimental.get\_device\_policy

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_device_policy#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_device_policy#aliases)

Gets the current device policy.

### Aliases:

* tf.compat.v1.config.experimental.get\_device\_policy
* tf.compat.v2.config.experimental.get\_device\_policy
* tf.config.experimental.get\_device\_policy

tf.config.experimental.get\_device\_policy()

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

The device policy controls how operations requiring inputs on a specific device (e.g., on GPU:0) handle inputs on a different device (e.g. GPU:1).

This function only gets the device policy for the current thread. Any subsequently started thread will again use the default policy.

#### Returns:

Current thread device policy

# tf.config.experimental.get\_memory\_growth

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_memory_growth#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_memory_growth#aliases)

Get if memory growth is enabled for a PhysicalDevice.

### Aliases:

* tf.compat.v1.config.experimental.get\_memory\_growth
* tf.compat.v2.config.experimental.get\_memory\_growth
* tf.config.experimental.get\_memory\_growth

tf.config.experimental.get\_memory\_growth(device)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

A PhysicalDevice with memory growth set will not allocate all memory on the device upfront.

#### For example:

physical\_devices = config.experimental.list\_physical\_devices('GPU')  
assert len(physical\_devices) > 0, "Not enough GPU hardware devices available"  
tf.config.experimental.set\_memory\_growth(physical\_devices[0], True)  
assert tf.config.experimental.get\_memory\_growth(physical\_devices[0]) == True

#### Args:

* **device**: PhysicalDevice to query

#### Returns:

Current memory growth setting.

# tf.config.experimental.get\_synchronous\_execution

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_synchronous_execution#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_synchronous_execution#aliases)

Gets whether operations are executed synchronously or asynchronously.

### Aliases:

* tf.compat.v1.config.experimental.get\_synchronous\_execution
* tf.compat.v2.config.experimental.get\_synchronous\_execution
* tf.config.experimental.get\_synchronous\_execution

tf.config.experimental.get\_synchronous\_execution()

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

TensorFlow can execute operations synchronously or asynchronously. If asynchronous execution is enabled, operations may return "non-ready" handles.

#### Returns:

Current thread execution mode

# tf.config.experimental.get\_virtual\_device\_configuration

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_virtual_device_configuration#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_virtual_device_configuration#aliases)

Get the virtual device configuration for a PhysicalDevice.

### Aliases:

* tf.compat.v1.config.experimental.get\_virtual\_device\_configuration
* tf.compat.v2.config.experimental.get\_virtual\_device\_configuration
* tf.config.experimental.get\_virtual\_device\_configuration

tf.config.experimental.get\_virtual\_device\_configuration(device)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Returns the list of VirtualDeviceConfiguration objects previously configured by a call to tf.config.experimental.set\_virtual\_device\_configuration().

#### For example:

physical\_devices = tf.config.experimental.list\_physical\_devices('CPU')  
assert len(physical\_devices) == 1, "No CPUs found"  
configs = tf.config.experimental.get\_virtual\_device\_configuration(  
    physical\_devices[0])  
assert configs is None  
tf.config.experimental.set\_virtual\_device\_configuration(  
    physical\_devices[0],  
    [tf.config.experimental.VirtualDeviceConfiguration(),  
     tf.config.experimental.VirtualDeviceConfiguration()])  
configs = tf.config.experimental.get\_virtual\_device\_configuration(  
    physical\_devices[0])  
assert len(configs) == 2

#### Args:

* **device**: PhysicalDevice to query

#### Returns:

List of [tf.config.experimental.VirtualDeviceConfiguration](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/VirtualDeviceConfiguration) objects or None if no virtual device configuration has been set for this physical device.

# tf.config.experimental.get\_visible\_devices

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_visible_devices#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/get_visible_devices#aliases)

Get the list of visible physical devices.

### Aliases:

* tf.compat.v1.config.experimental.get\_visible\_devices
* tf.compat.v2.config.experimental.get\_visible\_devices
* tf.config.experimental.get\_visible\_devices

tf.config.experimental.get\_visible\_devices(device\_type=None)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Returns a list of PhysicalDevice objects that are current marked as visible to the runtime. Any visible devices will have LogicalDevices assigned to them once the runtime is initialized.

The following example verifies all visible GPUs have been disabled:

physical\_devices = config.experimental.list\_physical\_devices('GPU')  
assert len(physical\_devices) > 0, "Not enough GPU hardware devices available"  
# Disable all GPUS  
tf.config.experimental.set\_visible\_devices([], 'GPU')  
visible\_devices = tf.config.experimental.get\_visible\_devices()  
for device in visible\_devices:  
  assert device.device\_type != 'GPU'

#### Args:

* **device\_type**: (optional) Device types to limit query to.

#### Returns:

List of PhysicalDevice objects

# tf.config.experimental.list\_logical\_devices

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/list_logical_devices#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/list_logical_devices#aliases)
* [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/list_logical_devices#used_in_the_guide)

Return a list of logical devices created by runtime.

### Aliases:

* tf.compat.v1.config.experimental.list\_logical\_devices
* tf.compat.v2.config.experimental.list\_logical\_devices
* tf.config.experimental.list\_logical\_devices

tf.config.experimental.list\_logical\_devices(device\_type=None)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

### Used in the guide:

* [Using GPUs](https://www.tensorflow.org/beta/guide/using_gpu)

Logical devices may correspond to physical devices or remote devices in the cluster. Operations and tensors may be placed on these devices by using the name of the LogicalDevice.

#### For example:

logical\_devices = tf.config.experimental.list\_logical\_devices('GPU')  
# Allocate on GPU:0  
with tf.device(logical\_devices[0].name):  
  one = tf.constant(1)  
# Allocate on GPU:1  
with tf.device(logical\_devices[1].name):  
  two = tf.constant(2)

#### Args:

* **device\_type**: (optional) Device type to filter by such as "CPU" or "GPU"

#### Returns:

List of LogicalDevice objects

# tf.config.experimental.list\_physical\_devices

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/list_physical_devices#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/list_physical_devices#aliases)
* [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/list_physical_devices#used_in_the_guide)

Return a list of physical devices visible to the runtime.

### Aliases:

* tf.compat.v1.config.experimental.list\_physical\_devices
* tf.compat.v2.config.experimental.list\_physical\_devices
* tf.config.experimental.list\_physical\_devices

tf.config.experimental.list\_physical\_devices(device\_type=None)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

### Used in the guide:

* [Using GPUs](https://www.tensorflow.org/beta/guide/using_gpu)

Physical devices are hardware devices locally present on the current machine. By default all discovered CPU and GPU devices are considered visible. The list\_physical\_devices allows querying the hardware prior to runtime initialization.

The following example ensures the machine can see at least 1 GPU.

physical\_devices = tf.config.experimental.list\_physical\_devices('GPU')  
assert len(physical\_devices) > 0, "No GPUs found."

#### Args:

* **device\_type**: (optional) Device type to filter by such as "CPU" or "GPU"

#### Returns:

List of PhysicalDevice objects

# tf.config.experimental.set\_device\_policy

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_device_policy#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_device_policy#aliases)

Sets the current thread device policy.

### Aliases:

* tf.compat.v1.config.experimental.set\_device\_policy
* tf.compat.v2.config.experimental.set\_device\_policy
* tf.config.experimental.set\_device\_policy

tf.config.experimental.set\_device\_policy(device\_policy)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

The device policy controls how operations requiring inputs on a specific device (e.g., on GPU:0) handle inputs on a different device (e.g. GPU:1).

When using the default, an appropriate policy will be picked automatically. The default policy may change over time.

This function only sets the device policy for the current thread. Any subsequently started thread will again use the default policy.

#### Args:

* **device\_policy**: A device policy. Valid values:
  + None: Switch to a system default.
  + 'warn': Copies the tensors which are not on the right device and logs a warning.
  + 'explicit': Raises an error if the placement is not as required.
  + 'silent': Silently copies the tensors. Note that this may hide performance problems as there is no notification provided when operations are blocked on the tensor being copied between devices.
  + 'silent\_for\_int32': silently copies int32 tensors, raising errors on the other ones.

#### Raises:

* **ValueError**: If an invalid device\_policy is passed.

# tf.config.experimental.set\_memory\_growth

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_memory_growth#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_memory_growth#aliases)
* [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_memory_growth#used_in_the_guide)

Set if memory growth should be enabled for a PhysicalDevice.

### Aliases:

* tf.compat.v1.config.experimental.set\_memory\_growth
* tf.compat.v2.config.experimental.set\_memory\_growth
* tf.config.experimental.set\_memory\_growth

tf.config.experimental.set\_memory\_growth(  
    device,  
    enable  
)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

### Used in the guide:

* [Using GPUs](https://www.tensorflow.org/beta/guide/using_gpu)

A PhysicalDevice with memory growth set will not allocate all memory on the device upfront. Memory growth cannot be configured on a PhysicalDevice with virtual devices configured.

#### For example:

physical\_devices = tf.config.experimental.list\_physical\_devices('GPU')  
assert len(physical\_devices) > 0, "Not enough GPU hardware devices available"  
tf.config.experimental.set\_memory\_growth(physical\_devices[0], True)

#### Args:

* **device**: PhysicalDevice to configure
* **enable**: Whether to enable or disable memory growth

# tf.config.experimental.set\_synchronous\_execution

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_synchronous_execution#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_synchronous_execution#aliases)

Specifies whether operations are executed synchronously or asynchronously.

### Aliases:

* tf.compat.v1.config.experimental.set\_synchronous\_execution
* tf.compat.v2.config.experimental.set\_synchronous\_execution
* tf.config.experimental.set\_synchronous\_execution

tf.config.experimental.set\_synchronous\_execution(enable)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

TensorFlow can execute operations synchronously or asynchronously. If asynchronous execution is enabled, operations may return "non-ready" handles.

When enable is set to None, an appropriate value will be picked automatically. The value picked may change between TensorFlow releases.

#### Args:

* **enable**: Whether operations should be dispatched synchronously. Valid values:
  + None: sets the system default.
  + True: executes each operation synchronously.
  + False: executes each operation asynchronously.

# tf.config.experimental.set\_virtual\_device\_configuration

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_virtual_device_configuration#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_virtual_device_configuration#aliases)
* [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_virtual_device_configuration#used_in_the_guide)

Set the virtual device configuration for a PhysicalDevice.

### Aliases:

* tf.compat.v1.config.experimental.set\_virtual\_device\_configuration
* tf.compat.v2.config.experimental.set\_virtual\_device\_configuration
* tf.config.experimental.set\_virtual\_device\_configuration

tf.config.experimental.set\_virtual\_device\_configuration(  
    device,  
    virtual\_devices  
)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

### Used in the guide:

* [Using GPUs](https://www.tensorflow.org/beta/guide/using_gpu)

A PhysicalDevice marked as visible will by default have a single LogicalDevice allocated to it once the runtime is configured. Specifying a list of tf.config.experimental.VirtualDeviceConfiguration objects allows multiple devices to be configured that utilize the same PhysicalDevice.

The following example splits the CPU into 2 virtual devices:

physical\_devices = tf.config.experimental.list\_physical\_devices('CPU')  
assert len(physical\_devices) == 1, "No CPUs found"  
# Specify 2 virtual CPUs. Note currently memory limit is not supported.  
tf.config.experimental.set\_virtual\_device\_configuration(  
  physical\_devices[0],  
  [tf.config.experimental.VirtualDeviceConfiguration(),  
   tf.config.experimental.VirtualDeviceConfiguration()])  
logical\_devices = tf.config.experimental.list\_logical\_devices('CPU')  
assert len(logical\_devices) == 2  
  
try:  
  tf.config.experimental.set\_virtual\_device\_configuration(  
    physical\_devices[0],  
    [tf.config.experimental.VirtualDeviceConfiguration(),  
     tf.config.experimental.VirtualDeviceConfiguration(),  
     tf.config.experimental.VirtualDeviceConfiguration(),  
     tf.config.experimental.VirtualDeviceConfiguration()])  
except:  
  print('Cannot modify the virtual devices once they have been initialized.')

The following example splits the GPU into 2 virtual devices with 100 MB each:

physical\_devices = tf.config.experimental.list\_physical\_devices('GPU')  
assert len(physical\_devices) > 0, "No GPUs found"  
tf.config.experimental.set\_virtual\_device\_configuration(  
  physical\_devices[0],  
  [tf.config.experimental.VirtualDeviceConfiguration(memory\_limit=100),  
   tf.config.experimental.VirtualDeviceConfiguration(memory\_limit=100)])  
  
try:  
  tf.config.experimental.set\_memory\_growth(physical\_devices[0], True)  
except:  
  print('Cannot set memory growth when virtual devices configured')  
  
logical\_devices = tf.config.experimental.list\_logical\_devices('GPU')  
assert len(logical\_devices) == len(physical\_devices) + 1  
  
try:  
  tf.config.experimental.set\_virtual\_device\_configuration(  
    physical\_devices[0],  
    [tf.config.experimental.VirtualDeviceConfiguration(memory\_limit=10),  
     tf.config.experimental.VirtualDeviceConfiguration(memory\_limit=10)])  
except:  
  print('Cannot modify the virtual devices once they have been initialized.')

#### Args:

* **device**: (optional) Need to update
* **virtual\_devices**: (optional) Need to update

# tf.config.experimental.set\_visible\_devices

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_visible_devices#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_visible_devices#aliases)
* [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/set_visible_devices#used_in_the_guide)

Set the list of visible devices.

### Aliases:

* tf.compat.v1.config.experimental.set\_visible\_devices
* tf.compat.v2.config.experimental.set\_visible\_devices
* tf.config.experimental.set\_visible\_devices

tf.config.experimental.set\_visible\_devices(  
    devices,  
    device\_type=None  
)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

### Used in the guide:

* [Using GPUs](https://www.tensorflow.org/beta/guide/using_gpu)

Sets the list of PhysicalDevices to be marked as visible to the runtime. Any devices that are not marked as visible means TensorFlow will not allocate memory on it and will not be able to place any operations on it as no LogicalDevice will be created on it. By default all discovered devices are marked as visible.

The following example demonstrates disabling the first GPU on the machine.

physical\_devices = config.experimental.list\_physical\_devices('GPU')  
assert len(physical\_devices) > 0, "Not enough GPU hardware devices available"  
# Disable first GPU  
tf.config.experimental.set\_visible\_devices(physical\_devices[1:], 'GPU')  
logical\_devices = config.experimental.list\_logical\_devices('GPU')  
# Logical device was not created for first GPU  
assert len(logical\_devices) == len(physical\_devices) - 1

#### Args:

* **devices**: (optional) List of PhysicalDevice objects to make visible
* **device\_type**: (optional) Device types to limit visibility configuration to. Other device types will be left unaltered.

# tf.config.experimental.VirtualDeviceConfiguration

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/VirtualDeviceConfiguration#top_of_page)
* [Class VirtualDeviceConfiguration](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/VirtualDeviceConfiguration#class_virtualdeviceconfiguration)
  + [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/VirtualDeviceConfiguration#aliases)
  + [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/VirtualDeviceConfiguration#used_in_the_guide)
* [Properties](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/VirtualDeviceConfiguration#properties)
  + [memory\_limit](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/experimental/VirtualDeviceConfiguration#memory_limit)

## Class VirtualDeviceConfiguration

Configuration class for virtual devices for a PhysicalDevice.

### Aliases:

* Class tf.compat.v1.config.experimental.VirtualDeviceConfiguration
* Class tf.compat.v2.config.experimental.VirtualDeviceConfiguration
* Class tf.config.experimental.VirtualDeviceConfiguration

Defined in [python/eager/context.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/eager/context.py).

### Used in the guide:

* [Using GPUs](https://www.tensorflow.org/beta/guide/using_gpu)

#### Fields:

* **memory\_limit**: (optional) Maximum memory (in MB) to allocate on the virtual device. Currently only supported for GPUs.

## Properties

### memory\_limit

Module: tf.config.optimizer

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer#top_of_page)
* [Functions](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer#functions)

Public API for tf.config.optimizer namespace.

Functions

[get\_experimental\_options(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/get_experimental_options): Get experimental optimizer options.

[get\_jit(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/get_jit): Get if JIT compilation is enabled.

[set\_experimental\_options(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/set_experimental_options): Set experimental optimizer options.

[set\_jit(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/set_jit): Set if JIT compilation is enabled.

# tf.config.optimizer.get\_experimental\_options

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/get_experimental_options#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/get_experimental_options#aliases)

Get experimental optimizer options.

### Aliases:

* tf.compat.v1.config.optimizer.get\_experimental\_options
* tf.compat.v2.config.optimizer.get\_experimental\_options
* tf.config.optimizer.get\_experimental\_options

tf.config.optimizer.get\_experimental\_options()

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Refer to tf.config.optimizer.set\_experimental\_options for a list of current options.

Note that optimizations are only applied in graph mode, (within tf.function). In addition, as these are experimental options, the list is subject to change.

#### Returns:

Dictionary of configured experimental optimizer options

# tf.config.optimizer.get\_jit

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/get_jit#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/get_jit#aliases)

Get if JIT compilation is enabled.

### Aliases:

* tf.compat.v1.config.optimizer.get\_jit
* tf.compat.v2.config.optimizer.get\_jit
* tf.config.optimizer.get\_jit

tf.config.optimizer.get\_jit()

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Note that optimizations are only applied in graph mode, (within tf.function).

#### Returns:

If JIT compilation is enabled.

# tf.config.optimizer.set\_experimental\_options

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/set_experimental_options#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/set_experimental_options#aliases)

Set experimental optimizer options.

### Aliases:

* tf.compat.v1.config.optimizer.set\_experimental\_options
* tf.compat.v2.config.optimizer.set\_experimental\_options
* tf.config.optimizer.set\_experimental\_options

tf.config.optimizer.set\_experimental\_options(options)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Note that optimizations are only applied in graph mode, (within tf.function). In addition, as these are experimental options, the list is subject to change.

#### Args:

* **options**: Dictionary of experimental optimizer options to configure. Valid keys:
  + layout\_optimizer: Optimize tensor layouts e.g. This will try to use NCHW layout on GPU which is faster.
  + constant\_folding: Fold constants Statically infer the value of tensors when possible, and materialize the result using constants.
  + shape\_optimization: Simplify computations made on shapes.
  + remapping: Remap subgraphs onto more efficient implementations.
  + arithmetic\_optimization: Simplify arithmetic ops with common sub-expression elimination and arithmetic simplification.
  + dependency\_optimization: Control dependency optimizations. Remove redundant control dependencies, which may enable other optimization. This optimizer is also essential for pruning Identity and NoOp nodes.
  + loop\_optimization: Loop optimizations.
  + function\_optimization: Function optimizations and inlining.
  + debug\_stripper: Strips debug-related nodes from the graph.
  + disable\_model\_pruning: Disable removal of unnecessary ops from the graph
  + scoped\_allocator\_optimization: Try to allocate some independent Op outputs contiguously in order to merge or eliminate downstream Ops.
  + pin\_to\_host\_optimization: Force small ops onto the CPU.
  + implementation\_selector: Enable the swap of kernel implementations based on the device placement.
  + auto\_mixed\_precision: Change certain float32 ops to float16 on Volta GPUs and above. Without the use of loss scaling, this can cause numerical underflow (seekeras.mixed\_precision.experimental.LossScaleOptimizer).
  + disable\_meta\_optimizer: Disable the entire meta optimizer.
  + min\_graph\_nodes: The minimum number of nodes in a graph to optimizer. For smaller graphs, optimization is skipped.

# tf.config.optimizer.set\_jit

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/set_jit#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/optimizer/set_jit#aliases)

Set if JIT compilation is enabled.

### Aliases:

* tf.compat.v1.config.optimizer.set\_jit
* tf.compat.v2.config.optimizer.set\_jit
* tf.config.optimizer.set\_jit

tf.config.optimizer.set\_jit(enabled)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

#### Args:

* **enabled**: Whether to enable JIT compilation.

Module: tf.config.threading

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading#top_of_page)
* [Functions](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading#functions)

Public API for tf.config.threading namespace.

Functions

[get\_inter\_op\_parallelism\_threads(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/get_inter_op_parallelism_threads): Get number of threads used for parallelism between independent operations.

[get\_intra\_op\_parallelism\_threads(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/get_intra_op_parallelism_threads): Get number of threads used within an individual op for parallelism.

[set\_inter\_op\_parallelism\_threads(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/set_inter_op_parallelism_threads): Set number of threads used for parallelism between independent operations.

[set\_intra\_op\_parallelism\_threads(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/set_intra_op_parallelism_threads): Set number of threads used within an individual op for parallelism.

# tf.config.threading.get\_inter\_op\_parallelism\_threads

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/get_inter_op_parallelism_threads#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/get_inter_op_parallelism_threads#aliases)

Get number of threads used for parallelism between independent operations.

### Aliases:

* tf.compat.v1.config.threading.get\_inter\_op\_parallelism\_threads
* tf.compat.v2.config.threading.get\_inter\_op\_parallelism\_threads
* tf.config.threading.get\_inter\_op\_parallelism\_threads

tf.config.threading.get\_inter\_op\_parallelism\_threads()

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Determines the number of threads used by independent non-blocking operations. 0 means the system picks an appropriate number.

#### Returns:

Number of parallel threads

# tf.config.threading.get\_intra\_op\_parallelism\_threads

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/get_intra_op_parallelism_threads#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/get_intra_op_parallelism_threads#aliases)

Get number of threads used within an individual op for parallelism.

### Aliases:

* tf.compat.v1.config.threading.get\_intra\_op\_parallelism\_threads
* tf.compat.v2.config.threading.get\_intra\_op\_parallelism\_threads
* tf.config.threading.get\_intra\_op\_parallelism\_threads

tf.config.threading.get\_intra\_op\_parallelism\_threads()

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Certain operations like matrix multiplication and reductions can utilize parallel threads for speed ups. A value of 0 means the system picks an appropriate number.

#### Returns:

Number of parallel threads

# tf.config.threading.set\_inter\_op\_parallelism\_threads

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/set_inter_op_parallelism_threads#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/set_inter_op_parallelism_threads#aliases)

Set number of threads used for parallelism between independent operations.

### Aliases:

* tf.compat.v1.config.threading.set\_inter\_op\_parallelism\_threads
* tf.compat.v2.config.threading.set\_inter\_op\_parallelism\_threads
* tf.config.threading.set\_inter\_op\_parallelism\_threads

tf.config.threading.set\_inter\_op\_parallelism\_threads(num\_threads)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Determines the number of threads used by independent non-blocking operations. 0 means the system picks an appropriate number.

#### Args:

* **num\_threads**: Number of parallel threads

# tf.config.threading.set\_intra\_op\_parallelism\_threads

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/set_intra_op_parallelism_threads#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/config/threading/set_intra_op_parallelism_threads#aliases)

Set number of threads used within an individual op for parallelism.

### Aliases:

* tf.compat.v1.config.threading.set\_intra\_op\_parallelism\_threads
* tf.compat.v2.config.threading.set\_intra\_op\_parallelism\_threads
* tf.config.threading.set\_intra\_op\_parallelism\_threads

tf.config.threading.set\_intra\_op\_parallelism\_threads(num\_threads)

Defined in [python/framework/config.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/framework/config.py).

Certain operations like matrix multiplication and reductions can utilize parallel threads for speed ups. A value of 0 means the system picks an appropriate number.

#### Args:

* **num\_threads**: Number of parallel threads