# tf.train.init_from_checkpoint

```
init_from_checkpoint(
    ckpt_dir_or_file,
    assignment_map
)
```

Defined in `tensorflow/python/training/checkpoint_utils.py`.

Initializes current variables with tensors loaded from given checkpoint.

**Note:** This overrides default initialization ops of specified variables and redefines dtype.

Assignment map supports following syntax:

- `'checkpoint_scope_name/': 'scope_name/'` - will load all variables in current `scope_name` from `checkpoint_scope_name` with matching tensor names.
- `'checkpoint_scope_name/some_other_variable': 'scope_name/variable_name'` - will initialize `scope_name/variable_name` variable from `checkpoint_scope_name/some_other_variable`.
- `'scope_variable_name': variable` - will initialize given `tf.Variable` object with tensor 'scope_variable_name' from the checkpoint.
- `'scope_variable_name': list(variable)` - will initialize list of partitioned variables with tensor 'scope_variable_name' from the checkpoint.
- `'/': 'scope_name/'` - will load all variables in current `scope_name` from checkpoint's root (e.g. no scope).

Supports loading into partitioned variables, which are represented as `'<variable>/part_<part #>'`.

Example:

```
# Say, '/tmp/model.ckpt' has the following tensors:
#  -- name='old_scope_1/var1', shape=[20, 2]
#  -- name='old_scope_1/var2', shape=[50, 4]
#  -- name='old_scope_2/var3', shape=[100, 100]

# Create new model's variables
with tf.variable_scope('new_scope_1'):
  var1 = tf.get_variable('var1', shape=[20, 2],
                         initializer=tf.zeros_initializer())
with tf.variable_scope('new_scope_2'):
  var2 = tf.get_variable('var2', shape=[50, 4],
                         initializer=tf.zeros_initializer())
  # Partition into 5 variables along the first axis.
  var3 = tf.get_variable(name='var3', shape=[100, 100],
                         initializer=tf.zeros_initializer(),
                         partitioner=lambda shape, dtype: [5, 1])

# Initialize all variables in `new_scope_1` from `old_scope_1`.
init_from_checkpoint('/tmp/model.ckpt', {'old_scope_1/', 'new_scope_1'})

# Use names to specify which variables to initialize from checkpoint.
init_from_checkpoint('/tmp/model.ckpt',
                     {'old_scope_1/var1': 'new_scope_1/var1',
                      'old_scope_1/var2': 'new_scope_2/var2'})

# Or use tf.Variable objects to identify what to initialize.
init_from_checkpoint('/tmp/model.ckpt',
                     {'old_scope_1/var1': var1,
                      'old_scope_1/var2': var2})

# Initialize partitioned variables using variable's name
init_from_checkpoint('/tmp/model.ckpt',
                     {'old_scope_2/var3': 'new_scope_2/var3'})

# Or specify the list of tf.Variable objects.
init_from_checkpoint('/tmp/model.ckpt',
                     {'old_scope_2/var3': var3._get_variable_list()})
```

## Args:

- `ckpt_dir_or_file` : Directory with checkpoints file or path to checkpoint.
- `assignment_map` : Dict, where keys are names of the variables in the checkpoint and values are current variables or names of current variables (in default graph).

## Raises:

- `tf.errors.OpError` : If missing checkpoints or tensors in checkpoints.
- `ValueError` : If missing variables in current graph.