Testing new Hugging Face Deep Learning Container.

This document explains the testing strategy for releasing the new Hugging Face Deep Learning Container. AWS maintains 14 days of currency with framework releases. Besides framework releases, AWS release train is bi-weekly on Monday. Code cutoff date for any changes is the Wednesday before release-Monday.

Test Case 1: Releasing a New Version (Minor/Major) of 🤗 Transformers

Requirements: Test should run on Release Candidate for new `transformers` release to validate the new release is compatible with the DLCs. To run these tests you need credentials for the HF SageMaker AWS Account. You can ask @philschmid or @n1t0 to get access.

Run Tests:

Before we can run the tests we need to adjust the `requirements.txt` for PyTorch under `/tests/sagemaker/scripts/pytorch` and for TensorFlow under `/tests/sagemaker/scripts/pytorch`. We adjust the branch to the new RC-tag.

```
git+https://github.com/huggingface/transformers.git@v4.5.0.rc0 # install main or adjust ist with vX.X.X for installing version specific-transforms
```

After we adjusted the `requirements.txt` we can run Amazon SageMaker tests with:

```
AWS_PROFILE=<enter-your-profile> make test-sagemaker
```

These tests take around 10-15 minutes to finish. Preferably make a screenshot of the successfully ran tests.

After Transformers Release:

After we have released the Release Candidate we need to create a PR at the Deep Learning Container Repository.

Creating the update PR:

1. Update the two latest `buildspec.yaml` config for PyTorch and TensorFlow. The two latest `buildspec.yaml` are the `buildspec.yaml` without a version tag and the one with the highest framework version, e.g. `buildspec-1-7-1.yml` and not `buildspec-1-6.yml`.

To update the `buildspec.yaml` we need to adjust either the `transformers_version` or the `datasets_version` or both. Example for upgrading to `transformers 4.5.0` and `datasets 1.6.0`.

```
account_id: &ACCOUNT_ID <set-$ACCOUNT_ID-in-environment>
region: &REGION <set-$REGION-in-environment>
base_framework: &BASE_FRAMEWORK pytorch
framework: &FRAMEWORK !join [ "huggingface_", *BASE_FRAMEWORK]
version: &VERSION 1.6.0
short_version: &SHORT_VERSION 1.6

repository_info:
  training_repository: &TRAINING_REPOSITORY
    image_type: &TRAINING_IMAGE_TYPE training
    root: !join [ "huggingface/", *BASE_FRAMEWORK, "/", *TRAINING_IMAGE_TYPE ]
```

```
    repository_name: &REPOSITORY_NAME !join ["pr", "-", "huggingface", "-",
*BASE_FRAMEWORK, "-", *TRAINING_IMAGE_TYPE]
    repository: &REPOSITORY !join [ *ACCOUNT_ID, .dkr.ecr., *REGION,
.amazonaws.com/,
      *REPOSITORY_NAME ]

images:
  BuildHuggingFacePytorchGpuPy37Cu110TrainingDockerImage:
    <<: *TRAINING_REPOSITORY
    build: &HUGGINGFACE_PYTORCH_GPU_TRAINING_PY3 false
    image_size_baseline: &IMAGE_SIZE_BASELINE 15000
    device_type: &DEVICE_TYPE gpu
    python_version: &DOCKER_PYTHON_VERSION py3
    tag_python_version: &TAG_PYTHON_VERSION py36
    cuda_version: &CUDA_VERSION cu110
    os_version: &OS_VERSION ubuntu18.04
    transformers_version: &TRANSFORMERS_VERSION 4.5.0 # this was adjusted from 4.4.2
to 4.5.0
    datasets_version: &DATASETS_VERSION 1.6.0 # this was adjusted from 1.5.0 to
1.6.0
    tag: !join [ *VERSION, '-', 'transformers', *TRANSFORMERS_VERSION, '-',
*DEVICE_TYPE, '-', *TAG_PYTHON_VERSION, '-',
      *CUDA_VERSION, '-', *OS_VERSION ]
    docker_file: !join [ docker/, *SHORT_VERSION, /, *DOCKER_PYTHON_VERSION, /,
      *CUDA_VERSION, /Dockerfile., *DEVICE_TYPE ]
```

2. In the PR comment describe what test, we ran and with which package versions. Here you can copy the table from Current Tests.
3. In the PR comment describe what test we ran and with which framework versions. Here you can copy the table from Current Tests. You can take a look at this PR, which information are needed.

Test Case 2: Releasing a New AWS Framework DLC

Execute Tests

Requirements:

AWS is going to release new DLCs for PyTorch and/or TensorFlow. The Tests should run on the new framework versions with current `transformers` release to validate the new framework release is compatible with the `transformers` version. To run these tests you need credentials for the HF SageMaker AWS Account. You can ask @philschmid or @n1t0 to get access. AWS will notify us with a new issue in the repository pointing to their framework upgrade PR.

Run Tests:

Before we can run the tests we need to adjust the `requirements.txt` for Pytorch under `/tests/sagemaker/scripts/pytorch` and for Tensorflow under `/tests/sagemaker/scripts/pytorch`. We add the new framework version to it.

```
torch==1.8.1 # for pytorch
tensorflow-gpu==2.5.0 # for tensorflow
```

After we adjusted the `requirements.txt` we can run Amazon SageMaker tests with.

```
AWS_PROFILE=<enter-your-profile> make test-sagemaker
```

These tests take around 10-15 minutes to finish. Preferably make a screenshot of the successfully ran tests.

After successful Tests:

After we have successfully run tests for the new framework version we need to create a PR at the Deep Learning Container Repository.

Creating the update PR:

1. Create a new `buildspec.yaml` config for PyTorch and TensorFlow and rename the old `buildspec.yaml` to `buildespec-x.x.x`, where `x.x.x` is the base framework version, e.g. if pytorch 1.6.0 is the latest version in `buildspec.yaml` the file should be renamed to `buildspec-yaml-1-6.yaml`.

To create the new `buildspec.yaml` we need to adjust the `version` and the `short_version`. Example for upgrading to `pytorch 1.7.1`.

```
account_id: &ACCOUNT_ID <set-$ACCOUNT_ID-in-environment>
region: &REGION <set-$REGION-in-environment>
base_framework: &BASE_FRAMEWORK pytorch
framework: &FRAMEWORK !join [ "huggingface_", *BASE_FRAMEWORK]
version: &VERSION 1.7.1 # this was adjusted from 1.6.0 to 1.7.1
short_version: &SHORT_VERSION 1.7 # this was adjusted from 1.6 to 1.7

repository_info:
  training_repository: &TRAINING_REPOSITORY
    image_type: &TRAINING_IMAGE_TYPE training
    root: !join [ "huggingface/", *BASE_FRAMEWORK, "/", *TRAINING_IMAGE_TYPE ]
    repository_name: &REPOSITORY_NAME !join ["pr", "-", "huggingface", "-",
*BASE_FRAMEWORK, "-", *TRAINING_IMAGE_TYPE]
    repository: &REPOSITORY !join [ *ACCOUNT_ID, .dkr.ecr., *REGION,
.amazonaws.com/,
      *REPOSITORY_NAME ]

images:
  BuildHuggingFacePytorchGpuPy37Cu110TrainingDockerImage:
    <<: *TRAINING_REPOSITORY
    build: &HUGGINGFACE_PYTORCH_GPU_TRAINING_PY3 false
    image_size_baseline: &IMAGE_SIZE_BASELINE 15000
    device_type: &DEVICE_TYPE gpu
    python_version: &DOCKER_PYTHON_VERSION py3
    tag_python_version: &TAG_PYTHON_VERSION py36
    cuda_version: &CUDA_VERSION cu110
    os_version: &OS_VERSION ubuntu18.04
```

```
    transformers_version: &TRANSFORMERS_VERSION 4.4.2
    datasets_version: &DATASETS_VERSION 1.5.0
    tag: !join [ *VERSION, '-', 'transformers', *TRANSFORMERS_VERSION, '-',
*DEVICE_TYPE, '-', *TAG_PYTHON_VERSION, '-',
      *CUDA_VERSION, '-', *OS_VERSION ]
    docker_file: !join [ docker/, *SHORT_VERSION, /, *DOCKER_PYTHON_VERSION, /,
      *CUDA_VERSION, /Dockerfile., *DEVICE_TYPE ]
```

2. In the PR comment describe what test we ran and with which framework versions. Here you can copy the table from Current Tests. You can take a look at this PR, which information are needed.

Current Tests

ID	Description	Platform	#GPUS	Collected & evaluated metrics
pytorch-transfromers-test-single	test bert finetuning using BERT fromtransformerlib+PT	SageMaker createTrainingJob	1	train_runtime, eval_accuracy & eval_loss
pytorch-transfromers-test-2-ddp	test bert finetuning using BERT from transformer lib+ PT DPP	SageMaker createTrainingJob	16	train_runtime, eval_accuracy & eval_loss
pytorch-transfromers-test-2-smd	test bert finetuning using BERT from transformer lib+ PT SM DDP	SageMaker createTrainingJob	16	train_runtime, eval_accuracy & eval_loss
pytorch-transfromers-test-1-smp	test roberta finetuning using BERT from transformer lib+ PT SM MP	SageMaker createTrainingJob	8	train_runtime, eval_accuracy & eval_loss
tensorflow-transfromers-test-single	Test bert finetuning using BERT from transformer lib+TF	SageMaker createTrainingJob	1	train_runtime, eval_accuracy & eval_loss
tensorflow-transfromers-test-2-smd	test bert finetuning using BERT from transformer lib+ TF SM DDP	SageMaker createTrainingJob	16	train_runtime, eval_accuracy & eval_loss