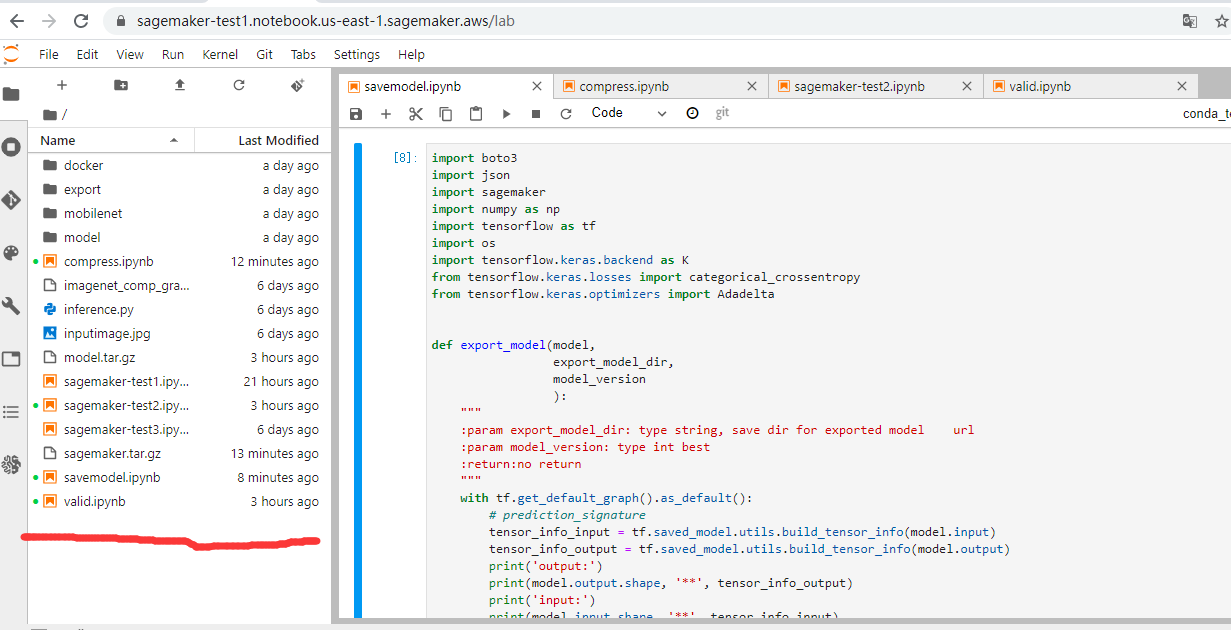
How to deploy pre-trained tensorflow model on Amazon sagemaker

The following steps are implemented by using sagemaker notebook (a kind of EC2 instance). Note that the notebook should be created with the authority to access the s3 storage. The file structures are shown as follows:



1. Prepare a pre-trained model. The pre-trained model and related serving code are packed into a compressed file named “model.tar.gz”.
2. Create a file folder ”export/Servo/code” in current directory;
3. Copy “inference.py” into “export/Servo/code”. The “inference.py” is used to process the input and output messages of serving-request.
4. Run “savemodel.ipynb” to save a pre-trained model in “export/Servo” and pack folder “export” into “model.tar.gz”. Then the structure of the folder “export” can be shown as：

-model.tar.gz

|--export

|--Servo

|--code

|--inference.py

|--1

|--saved\_model.pb

|--variables

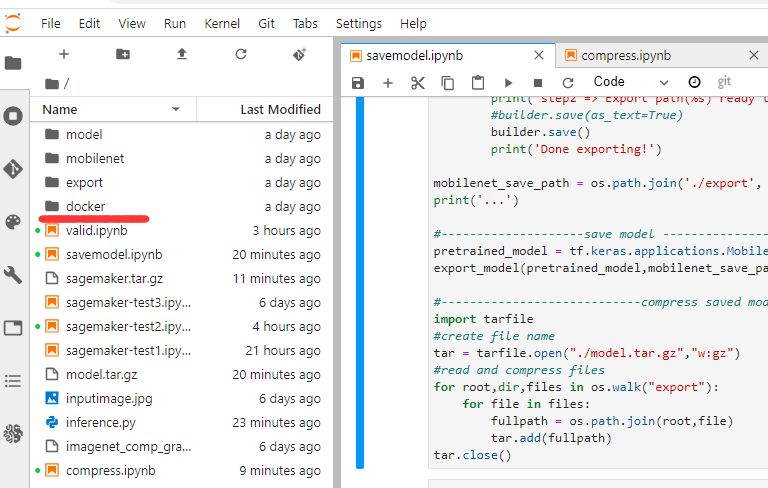
|--variables.index

|--variables.data-00000-of-00001

1. Upload “model.tar.gz” into amazon s3 storage, e.g., the directory on amazon s3 is “s3://mysagemaker-us-east/sagemaker\_model/model.tar.gz”



1. Prepare docker image for running the pre-trained model. The tensorflow version of docker images should be the same as that of the saved model, e.g, tensorflow 1.15 in our example.
2. Create a folder “docker” in current directory



1. Open a “terminal” of the sagemaker notebook and clone “Dockerfile” for building docker image. The commands are shown as follows:

$ cd SageMaker/docker

$ git clone <https://github.com/aws/sagemaker-tensorflow-serving-container.git>

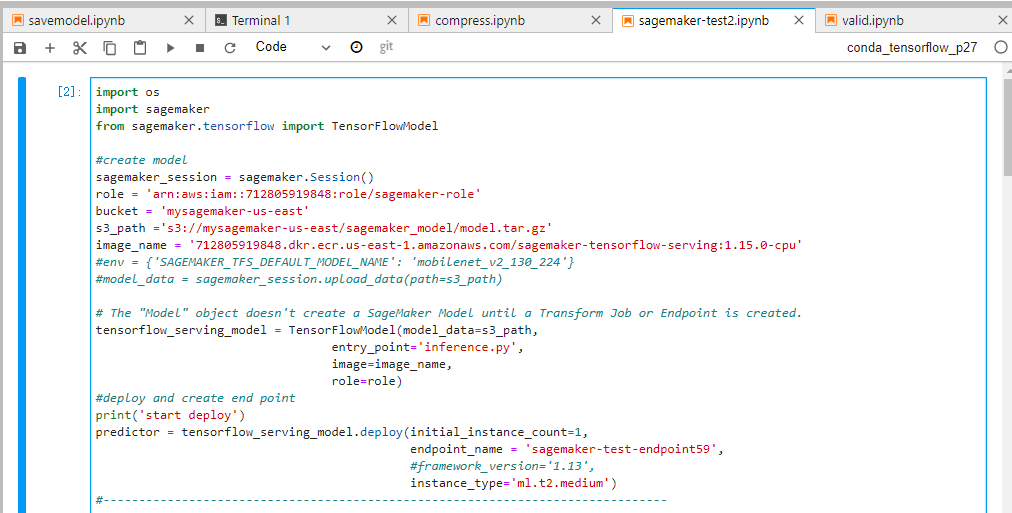
$ export AWS\_DEFAULT\_REGION=us-east-1 #set default region

$ ./scripts/build.sh --version 1.15 --arch cpu #build docker image

$ ./scripts/publish.sh --version 1.15 --arch cpu #push to amazon ECR storage.

#if push failed, maybe you should create a directory in your ECR, e.g., “sagemaker-tensorflow-serving”, and tray again.

1. Deploy pre-trained model by using the built docker image. The example code is shown in “sagemaker-test2.ipynb” where we should configure the model path in amazon s3 storage and docker image we pushed into amazon ECR.



1. Use the code “valid.ipynb” to test the deployed model.

