Distributed TensorFlow Mask-RCNN training

Step-by-step instructions

These instructions are to be followed step-by-step, mechanically. There will be a session during the Immersion Day that will explain what is happening at each step. This is needed because model training takes several hours, yet it must be completed before we can run inference on the model.

# Set up the environment

## Do this first

1. Open a terminal window and navigate to a folder where you can create a git repo.
2. Clone the sagemaker examples repo from git:  
   git clone https://github.com/awslabs/amazon-sagemaker-examples.git
3. Go into the repo:  
   cd amazon-sagemaker-examples
4. Checkout the maskrcnn\_inference branch:  
   git checkout maskrcnn\_inference
5. Follow one of 1.2 or 1.3 bellow.

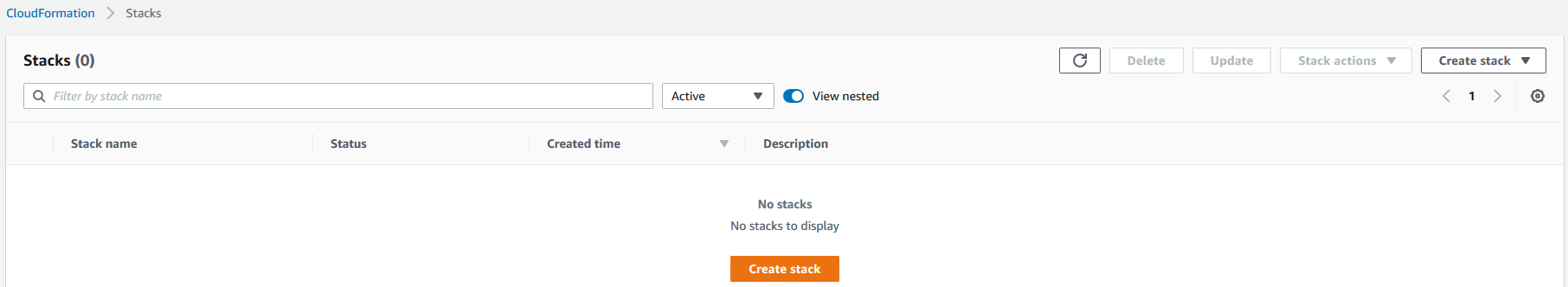
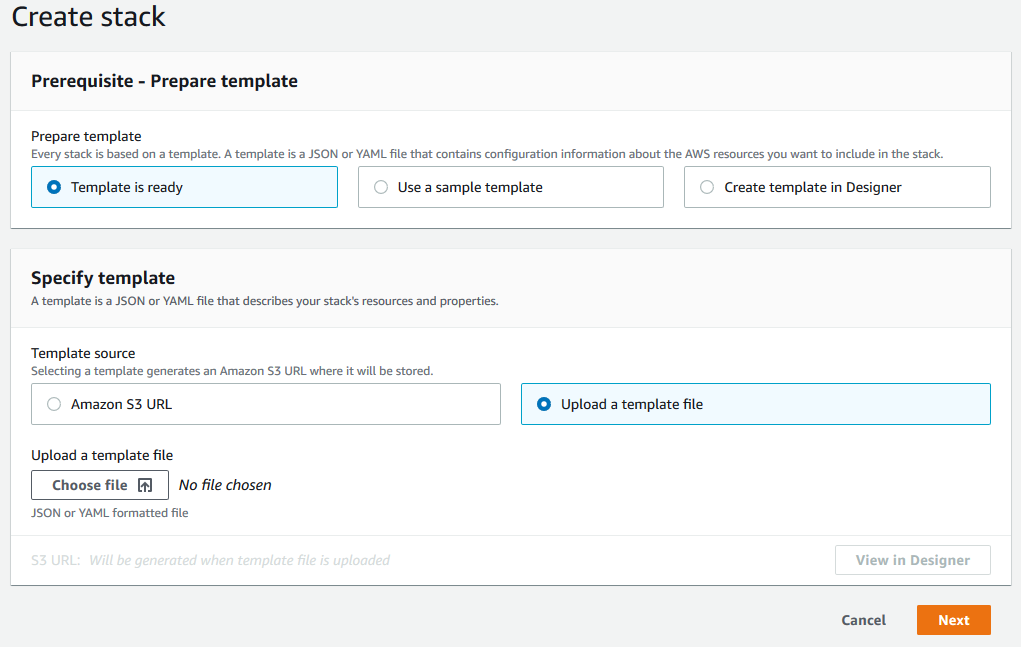
## Via Command Line

1. Navigate to the folder containing the lab code:  
   cd advanced\_functionality/distributed\_tensorflow\_mask\_rcnn
2. Open the stack-sm.sh file in a text editor of your preference
3. Make the following edits:

* On line 4, set AWS\_REGION="us-east-1"
* On line 8, set S3\_BUCKET="mask-rcnn-demo-coco-dataset-iad"

1. (Optional) On line 17, customize the name of the notebook that’ll be created. The default is NOTEBOOK\_INSTANCE\_NAME="sm-nb-$DATE"
2. The variable EFS\_ID on line 25 will be intentionally left blank. That will create a new Elastic File System that we’ll populate with a copy of the dataset to speed up training data serving.
3. The variables GIT\_URL, GIT\_USER and GIT\_TOKEN can be left blank for now. If you want to keep a copy of the work on the notebook on a personal repo, please set them to proper values.
4. Save the edited script and run it.
5. You will get a report on the shell of the stack creation progress and a (long) listing of the created stack at the end. You can also see the creation and its results via the AWS Console.

## Via Amazon Console

1. Login to the AWS Account provided
2. On the top right, make sure you are on region "us-east-1". If needed, change it to us-east-1 using the dropdown.
3. On the “Find Services” Search box, start typing “CloudFormation”. When it show up on the drop-down box, click on it.
4. On the CloudFormation main window, click on the “Create Stack” button on the **top right** and select “With new resources”.  
   
5. On the “Create Stack” page, select “Template is ready” under “Prepare template” and “upload a template file” under “Specify template”.  
   
6. Click on “Choose file”, navigate to the git repo you cloned before, go to the amazon-sagemaker-examples\advanced\_functionality\distributed\_tensorflow\_mask\_rcnn folder and select the cfn-sm.yaml file.
7. Click “Next”.
8. On the “Specify Stack Details” page, give the stack a name. We suggest “sm-stack-2020-03-10”.
9. Under “Parameters”, go to “S3BucketName” and fill it with "mask-rcnn-demo-coco-dataset-iad". Leave the other parameters as they are.
10. Click “Next”.
11. Under “Configure stack options”, scroll to the bottom and click “Next”.
12. Under “Review <stack-name>”, scroll to the bottom, acknowledge the “**I acknowledge that AWS CloudFormation might create IAM resources.” checkbox and click on “Create stack”.**
13. **You will be taken back to the list of stacks.**

## **Monitoring Stack Creation.**

1. **If you’re not on the list of stacks, follow steps 1 to 3 on the previous section.**
2. **On the list of stacks, your stack should be showing as “CREATE\_IN\_PROGRESS”, in blue. Click on it.**
3. **On the Stack details page, you can inspect the events, resources and outputs generated by the CloudFormation template.**
4. **Leave this tab open, we’ll need it moving forward (you can clone it or copy/paste the URl on a new tab to continue with the rest of the lab).**

# Access the Notebook

1. Navigate to SageMaker
2. Click on Notebook Instances
3. Locate the Notebook Instance you have created. It should be the only one In Service.
4. Click on Open JupyterLab.
5. I should have cloned the repo here. Better yet, create one with just the lab and clone it here.
6. On the left side, click on the folder icon.
7. Navigate to amazon-sagemaker-examples/advanced\_functionality/distributed\_tensorflow\_mask\_rcnn
8. Open mask-rcnn-efs.ipynb

# Execute the Notebook

1. Set region and bucket as you did on the cloudformation
2. Execute the cells to stage COCO 2017 dataset in Amazon S3 and to copy COCO 2017 dataset from S3 to Amazon EFS
3. Keep executing cells until you get to one containing the following command:  
   %%time  
   ! ./container/build\_tools/build\_and\_push.sh {aws\_region}
4. Execute the command, wait for it to finish, scroll to the end of the cell output and copy the Amazon ECR URI. It should look like this:  
   999999999999.dkr.ecr.us-east-1.amazonaws.com/mask-rcnn-tensorpack-sagemaker:tf1.13-xxxxxxxxx
5. On the next cell, replace #<amazon-ecr-uri> with the URI you just copied, within quotations.
6. Skip the cells under “AWS Samples Mask R-CNN” and go directly to the first cell under “SageMaker Initialization”
7. Execute that cell. On the next one, replace # set to tensorpack\_image or aws\_samples\_image with tensorpack\_image. Execute the cell after the edit.
8. Under “Define SageMaker Data Channels”:

* The first cell will retrieve the id of the EFS attached to the instance. That will be needed in the second cell.
* In the second cell, replace # 'fs-xxxxxxxx' with the value displayed in the first cell output, in quotations. Then run the cell.
* Run the rest of the cells in the section.

1. Run all cells in the “Configure Hyper-parameters” and “Define Training Metrics” sections.
2. On the “Define SageMaker Training Job” Section:

* Go back to the CloudFormation tab, open the details of the stack created earlier and select the “Outputs tab”
* Copy the “SecurityGroup” value and, in the first cell of the notebook, replace # ['sg-xxxxxxxx'] with it (including square brackets and quotations).
* Back in the CloudFormation tab, copy the “SubNets” value and replace # [ 'subnet-xxxxxxx'] with it (again including square brackets and quotations).
* Execute the two cells.
* Monitor the output of the mask\_rcnn\_estimator.fit(inputs=data\_channels, logs=True) cell for about 10 minutes. There should be a long output in different colors – each color represents a different host. There will be some warnings and information messages (repeated in each color), but at some point there should be an output like this:

--------Begin MPI Run Command----------

HOROVOD\_CYCLE\_TIME=0.5 \

HOROVOD\_FUSION\_THRESHOLD=67108864 \

mpirun -np 32 \

.

.

.

--------End MPI Run Comamnd------------

Data for JOB [63755,1] offset 0 Total slots allocated 32

======================== JOB MAP ========================

Data for node: ip-172-30-0-88#011Num slots: 8#011Max slots: 0#011Num procs: 8

Process OMPI jobid: [63755,1] App: 0 Process rank: 0 Bound: N/A

.

.

.

* You can go back to the console, on the SageMaker page (click on Services, select “SageMaker” from the list on the left side) and monitor the training job from there. On the SageMaker console page, click on “Training Jobs” on the left menu bar and locate the running training job (it should have the name of the notebook and a timestamp) and click on it. Scrolling down to the “Monitor” section will show you logs of the machines in use (CPU, Memory, GPU, etc.). The “View Algorithm Metrics” link will take you to CloudWatch Metrics, where all algorithm metrics will be displayed. This has been configured to log per epoch, so it is probably empty at this point.

Congratulations! You got the Mask-RCNN model training going.